User Manual

PXA125

125 MS/s

ARBITRARY WAVEFORM GENERATOR

Publication No. 010520

PUBLICATION DATE: June, 2002

REVISION: 1.1

WARRANTY STATEMENT

Products sold by LeCroy are warranted to be free from defects in workmanship or materials. LeCroy will, at its option, either repair or replace any hardware products, which prove to be defective during the warranty period. You are a valued customer. Our mission is to make any necessary repairs in a reliable and timely manner.

Duration of Warranty

The warranty period for this LeCroy hardware is one year, except software and firmware products designed for use with LeCroy Hardware is warranted not to fail to execute its programming instructions due to defect in materials or workmanship for a period of ninety (90) days from the date of delivery to the initial end user.

Return of Product

Authorization is required from LeCroy before you send your product for service or calibration. Call your nearest LeCroy support facility. A list is located on the last page of this manual. If you are unsure where to call, contact LeCroy Customer Support Department.

Limitation of Warranty

LeCroy shall be released from all obligations under this warranty in the event repairs or modifications are made by persons other than authorized LeCroy service personnel or without the written consent of LeCroy.

LeCroy expressly disclaims any liability to its customers, dealers and representatives and to users of its product, and to any other person or persons, for special or consequential damages of any kind and from any cause whatsoever arising out of or in any way connected with the manufacture, sale, handling, repair, maintenance, replacement or use of said products.

Representations and warranties made by any person including dealers and representatives of LeCroy, which are inconsistent or in conflict with the terms of this warranty (including but not limited to the limitations of the liability of LeCroy as set forth above), shall not be binding upon LeCroy unless reduced to writing and approved by an officer of LeCroy

Except as stated above, LeCroy makes no warranty, express or implied (either in fact or by operation of law), statutory or otherwise; and except to the extent stated above, LeCroy shall have no liability under any warranty, express or implied (either in fact or by operation of law), statutory or otherwise.

PROPRIETARY NOTICE

Table of Contents

Chapter 1 - PORTRAYAL

Chapter 2 - INSTALLATION

Chapter 3 - ArbConnection

List of Figures

List of Tables

This page intentionally left blank

Chapter 1
PORTRAYAL

What's In This Chapter

This chapter contains general and functional description of the Model PXA125 Arbitrary Waveform Generator. It also describes the front panel connectors and operational modes and provides description of all features available with the instruments.

Introduction

Model PXA125 is a single-channel PXI-based Arbitrary Waveform Generator. It is a high performance waveform generator that combines three powerful instruments in one small package: function generator, Waveform generator and modulation generator. Supplied free with the instrument is ArbConnection software, which is used for controlling the PXA125 and for generating, editing and downloading waveforms from a remote computer. The following highlights the PXA125 and ArbConnection features.

Conventions Used in this Manual

The following conventions may appear in this manual:

NOTE
A Note contains information relating to the use of this product

CAUTION
A Caution contains information that should be followed to avoid personal damage to the instrument or the equipment connected to it.

WARNING
A Warning alerts you to a potential hazard. Failure to adhere to the statement in a WARNING message could result in personal injury.

The following symbol may appear on the product:

⚠ ***CAUTION:***
Refer to Accompanying Documents

This refers you to additional information contained in this manual. The corresponding information in the manual is similarly denoted.

PXA125 Series

- PXI Single Slot Module

Feature Highlights

- 125 MS/s sample clock frequency
- Built-in standard waveforms. Up to 50 MHz sinewave output
- 10 digits frequency setting, limited by 1 µS/s
- 14-bit vertical resolution
- 2 Meg memory depth
- Ultra fast waveform downloads
- 1 ppm clock stability
- Extremely low phase noise carrier
- Frequency agility: FSK, ramped FSK, sweep, FM
- Trigger start phase control and breakpoints
- Built-in sequence generator
- Multiple instrument synchronization with tight phase control

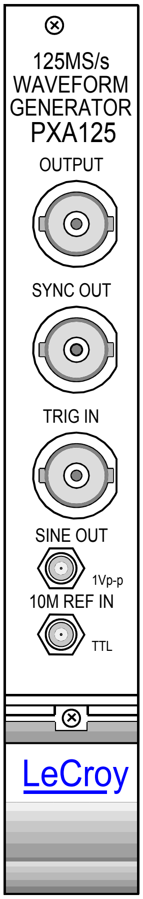


Figure 1-1, PXA125

ArbConnection Feature Highlights

- Three powerful tools in one software package: Instrument control panel, Waveform composer and FM signal composer
- Detailed virtual front panels control all PXA125 functions and modes
- Wave composer generates, edits and downloads complex waveforms
- FM wave composer generates and downloads complex modulating signals
- Automatic detection of active instruments
- Equation editor generates waveforms from equations
- SCPI command and response editor simulates ATE operation

- Translates waveform coordinates from ASCII and other formats
- Simplifies generation of complex sequences

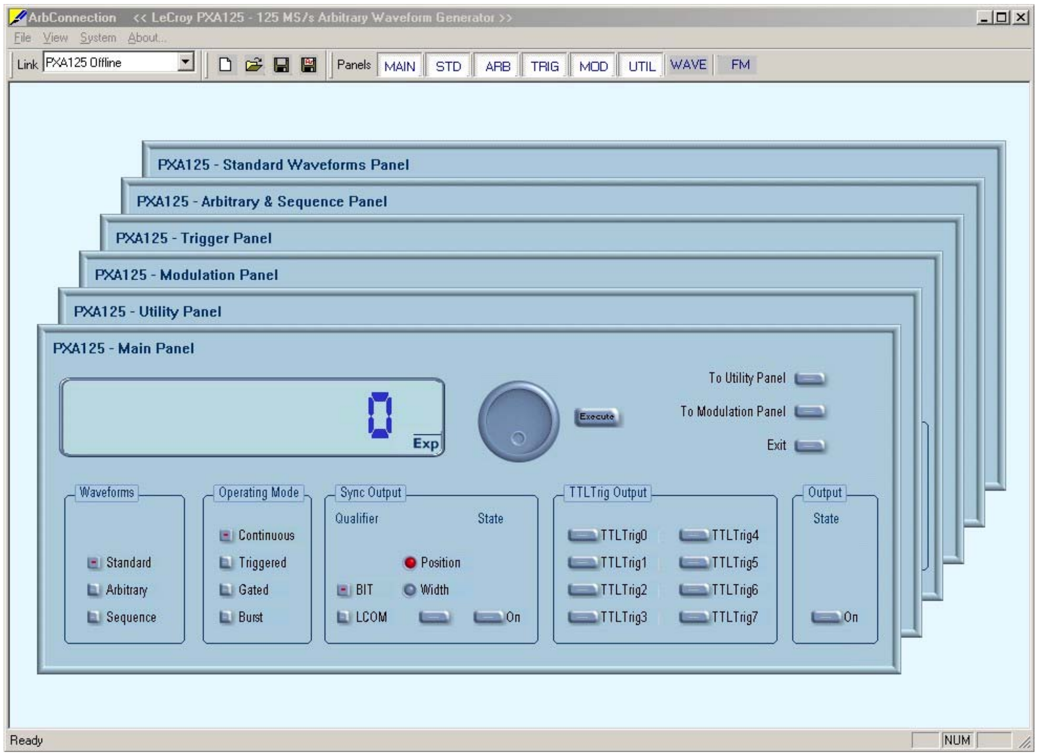
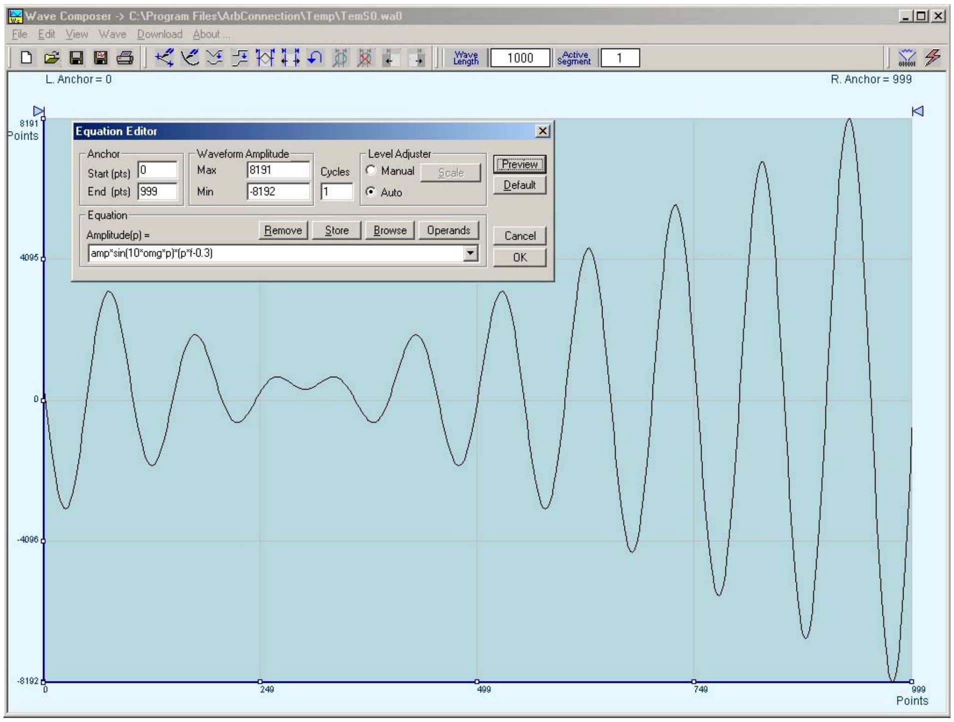


Figure 1-2, ArbConnection - Control Panels



Wave Composer -> C:\Program Files\ArbConnection\Temp\TemS0.wa0

Wave Length 1000 Active Segment 1

Anchor: Start (pts) 0, End (pts) 999

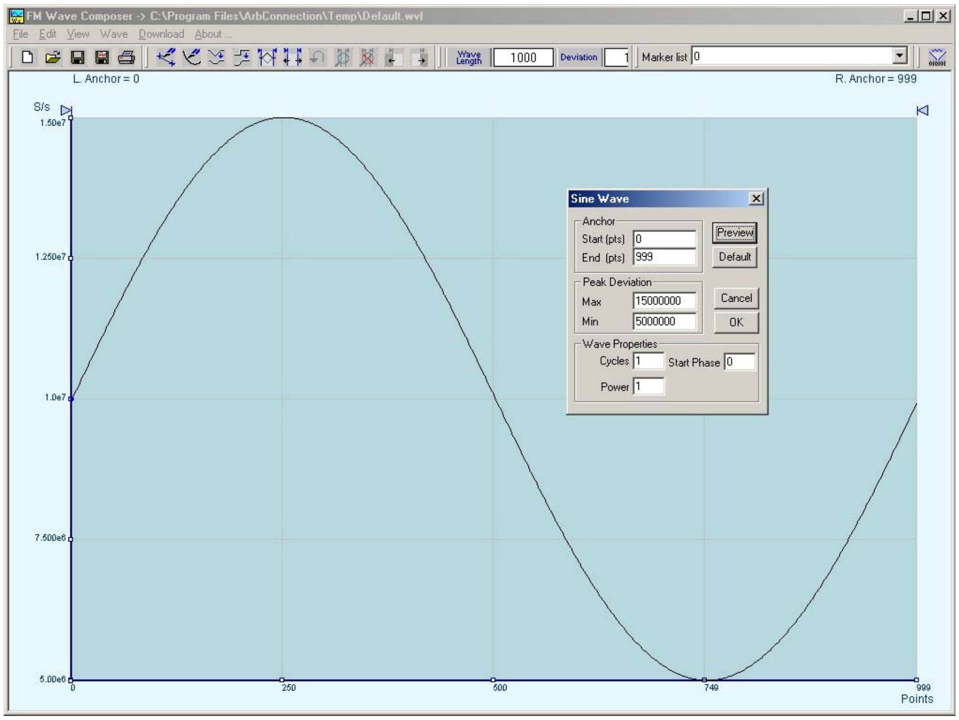
Waveform Amplitude: Max 8191, Min -8192

Cycles 1

Level Adjuster: ○ Manual ● Auto

Equation: Amplitude(p) = amp*sin(10*omg*p)*(p*f-0.3)

Figure 1-3, ArbConnection - Wave Composer



FM Wave Composer -> C:\Program Files\ArbConnection\Temp\Default.wvf

Wave Length 1000 Deviation 1 Marker list 0

Anchor: Start (pts) 0, End (pts) 999

Peak Deviation: Max 15000000, Min 5000000

Wave Properties: Cycles 1, Start Phase 0, Power 1

Figure 1-4, ArbConnection - FM Wave Composer

Functional Description

Detailed functional description is given following the general description of the features and functions available with the PXA125.

Output Functions

Model PXA125 is completely digital. There are no analog functions resident in its hardware circuits. Data has to be downloaded to the instrument for it to start generating waveforms. The instrument can generate a few standard functions such as sine wave, triangular wave and square wave. Each time that a standard function is required, the instrument calculates its coordinates and places them in the waveform memory. Therefore, every time a standard function is selected, minimal time is required for the controller to compute the function and load its data to the waveform memory.

Frequency

Waveform frequency and sample clock are programmed with 10 digits, limited only by 1 μS/s. Frequency accuracy of the output waveform is determined by the clock reference, CLK10, which provides 100ppm accuracy and stability over time and temperature.

Amplitude

The output level may be programmed from 160 mV to 16 Vp-p into open circuit, or 80 mV to 8 V into 50Ω. Offset may be applied to the output to shift the signal either positive or negative. Offset and amplitude are inter-related, so make sure you understand the offset-amplitude ranges before you apply offset to your signal.

Trigger Modes

Besides its normal continuous mode, the PXA125 responds to a variety of trigger sources. The output waveform may be gated, triggered, or generate a counted burst of waveforms. A built-in trigger generator, having a programmable period can be used as a replacement of an external trigger source. The internal trigger generator can be programmed with resolution of 7 digits. The PXA125 can be programmed to output triggers on one or more of 8 trigger lines that are connected to P2, as well as, be programmed to respond to these same trigger lines.

Arbitrary Waveforms

The Model PXA125 generates arbitrary waveforms with 14 bits of vertical resolution. Any waveform it generates must first be loaded to its waveform memory. The arbitrary waveform memory is a bank of 14-bit words. Each word represents a point on the horizontal waveform scale. Each word has a horizontal address that can range from 0 to 2,097,152 and a vertical address that can range from -8192 to +8191 (14 bits). Using a high speed clocking circuit, the digital contents of the arbitrary waveform memory is extracted and routed to the Digital to Analog Converter (DAC). The DAC converts the digital data to an analog signal, and the output amplifier completes the task by amplifying or attenuating the signal at the output connector.

Memory Segmentation

There is no need to use the entire memory every time an arbitrary waveform is generated. The waveform memory can be divided into smaller segments and different waveforms can be loaded into individual segment. The various segments may then be loaded into a sequence table to generate long and complex waveforms. The sequence table can link up to 4096 segments, while each segment can loop up to 128 K times.

Remote Control The instrument must be used in conjunction with a host computer. All of its functions, modes and parameters are fully programmable using SCPI commands and syntax. There are three ways to program the Model PXA125, the first being low-level programming of each individual parameter, using SCPI commands. The second alternative is to use ArbConnection for high-level programming. ArbConnection is a software package supplied with the PXA125 that simulates a mechanical front panel. It has all the necessary push buttons, displays and dials to operate the instrument as if you were using it on the bench. The third alternative is using application specific drivers, such LabVIEW.

The PXA125 must be programmed to generate waveforms. Therefore, it is recommended that the user becomes familiar with its basic features, functions and programming concepts as described in this and subsequent chapters.

Frequency Agility The instrument generates its sample clock from a DDS circuit (direct digital synthesis). The DDS circuit enables frequency agility through the complete frequency range of the PXA125. Having such an enormous range opens the door for a wide range of applications such as wide band sweep, FSK and frequency modulation. The PXA125 can generate FSK, Ramped FSK and Linear or Logarithmic sweep. The instrument can also frequency modulate its carrier using one of its built-in waveforms, or with can also generate any user-defined modulating signal, which can be downloaded using the FM wave composer.

Multi-Instrument Synchronization There are applications requiring 2 or more synchronized channels. Synchronization between completely independent, free-running instruments is not an easy task. Besides distribution of the sample clock to all instruments, each instrument has to be told when to start generating waveforms so that all waveforms start from the same point and with the same phase. Also, triggering multiple instruments to achieve synchronization is not enough because it will generate a jitter of ±1 count.

The PXA125 is using Backplane, daisy-chained connections to synchronize between modules and the only limitation of the number of synchronized instruments is determined by the size of the cage.

Supplied

The instrument is supplied with a CD that includes an Instruction

Accessories

Manual, ArbConnection for Windows 95/98/ME/2000/XP/NT and plug & play drivers.

Specifications

Instrument specifications are listed in Appendix A. These specifications are the performance standards or limits against which the instrument is tested. Specifications apply under the following conditions: output terminated into 50Ω after 30 minutes of warm up time, and within a temperature range of 20°C to 30°C. Specifications outside this range are degraded by 0.1% per °C.

Functional Description

A detailed functional description is given in the following paragraphs. The description is divided into logical groups: front panel connectors, operating modes, output type, output state, filters and synchronization.

Front Panel Connectors

The PXA125 has 3 BNC connectors on its front panel: main and SYNC outputs and trigger input. There are also 2 SMB connectors: sine output and 10 MHz reference input. These connectors are described below.

Output

The output connector outputs fixed (pre-defined) waveforms to 50 MHz, user (arbitrary) and sequenced waveforms with sampling clock to 125 MS/s. Output impedance is 50Ω, that is, the cable connected to this output should be terminated with 50Ω load. Amplitude accuracy is calibrated when connected to a 50Ω load. The amplitude is doubled when the output impedance is above 1 MΩ.

SYNC Output

The SYNC output generates a single TTL pulse for synchronizing other instruments (i.e., an oscilloscope) to the output waveform. The SYNC signal always appears at a fixed point relative to the waveform. The location of the SYNC signal along the waveform is programmable. The SYNC output is also used as marker output when the sweep, or FM functions are turned on.

TRIG/FSK IN

In general, this input accepts signals that stimulate generation of output waveforms. The trigger input is inactive when the generator operates in continuous mode. When placed in trigger, gated or burst modes, the trigger input is made active and waits for the right condition to trigger the instrument. In trigger and burst modes, the trigger input is edge sensitive, i.e., it senses transitions from high to low or from low to high to trigger the PXA125. The direction of the transition is programmable. In gated mode, the trigger input is level sensitive, i.e., the generator is gated when the logic level is high and idle when the level is logic low. Trigger level for this input is TTL.

The same input is used in FSK mode, where the output hops between two frequencies – carrier and shifted frequencies. The output generates carrier frequency when the FSK input is false and shifted frequency when the FSK input is true.

The trigger input is also used as stop and start input when the PXA125 is placed in Arm mode.

SINE OUT

This SMB connector outputs dc coupled, fixed level (1 V_{pk-pk} into 50Ω) sine waveforms. This output is derived directly from the sample clock generator and is active at all times, regardless of present operating mode of the PXA125. The frequency of the sine output is programmed using the sample clock parameter. Frequency agility and modulation affect this output directly. The sine waveform output is programmed to 125 MHz, so it may serve as an additional output to those available on the front panel.

REF IN

This SMB connector accepts 10 MHz, TTL level reference signal. The external reference input is available for those applications requiring better accuracy and stability reference than the one provided inside the PXA125. The reference input is active only after selecting the external reference source mode.

Operating Modes

The PXA125 can be programmed to operate in one of four operating modes: continuous, triggered, gated and (counted) burst. These modes are described below.

Continuous Mode

In normal continuous mode, the selected waveform is generated continuously at the selected frequency, amplitude and offset.

Triggered Mode

In triggered mode, the PXA125 circuits are armed to generate one output waveform. The trigger circuit is sensitive to transitions at the trigger input. Select between positive or negative transitions to trigger the instrument. When triggered, the generator outputs one waveform cycle and remains idle at the first point of the waveform. The instrument can be armed to receive a trigger signal from either the front panel connector, soft command, an internal trigger generator, from one of 8 backplane trigger lines or from a backplane STAR line. As an alternative to an external source, the PXA125 has a built-in trigger generator that can be programmed through a wide range of frequencies and with 7 digits of resolution.

The trigger signal, whether it comes from the front panel, backplane or from a soft command, has to pass through some electrical circuits. These circuits cause small delay known as system delay. System delay cannot be eliminated completely and must be considered when applying a trigger signal. It defines how long it will take from a valid trigger edge to the moment that the output reacts.

Gated Mode

In gated mode, the PXA125 circuits are armed to generate output waveforms as long as a gating signal is present. Unlike the triggered mode, the gated mode is level sensitive. When the gating signal goes low, the waveform at the output connector is first completed and the output reverts to an idle state. The idle amplitude level, after the gating signal goes low, is the last point on the waveform.

Burst Mode

The burst mode is an extension of the triggered mode where the generator can be programmed to output a pre-determined number of waveforms. The sources to trigger a burst are the same as for the trigger mode.

Frequency Agility

Using the latest DDS (direct digital synthesis) technology, the PXA125 is extremely agile. Operations like sweep, FSK and FM are directly derived from the DDS circuit by controlling its input bits. Frequency agility is described below.

Sweep

The PXA125 can sweep from minimum to maximum sample clock frequency settings. You may select to sweep up or down using linear or logarithmic increments. Sweep time is programmable from 1 ms to 1000 seconds with 7 digits. The sweep start and stop frequencies program the front-panel sine output connector. You may also use the sweep from the main output but must first calculate start stop frequencies, depending on the present sample clock frequency and waveform length. You may use the sweep in continuous, triggered, or gated modes.

FSK

FSK (frequency shift keying) function controls the sine output connector. The trigger input is used to flag the PXA125 when to output carrier frequency (trigger false) or when it should switch to the shift frequency (trigger true). You may also use the FSK function from the front panel as long as you do your own calculation of carrier and shifted frequencies, depending on the present sample clock frequency and waveform length.

Ramped FSK

Ramped FSK is the same as the FSK function except the output frequency is ramped instead of switched to the shift frequency. Ramp time is programmable with 3 digits from 100 μs to 1 second.

FM

The FM function modulates the PXA125 sample clock frequency. You can frequency modulate the output either with built-in waveforms, or download complex waveforms to the modulation memory. Using the latest DDS technology, the modulation is wide band and extremely linear. FM can be used in continuous, triggered and gated modes. The PXA125 can generate two types of frequency modulation: 1) Standard and 2) Arbitrary. In standard mode, the modulating waveform is selected from a built-in library of 4 standard waveforms: sine, triangle, square and ramped. In arbitrary mode, the modulating signal is downloaded to the modulation waveform memory. There are 20,000 points allocated specifically for the arbitrary memory that is used for arbitrary frequency modulation.

Output Type

The PXA125 can output three types of waveforms: standard (Fixed), arbitrary (User) and sequenced waveforms. Description of the various waveform types that the instrument can generate is given below.

Standard (FIXED) Waveforms

The PXA125 must pre-load its memory before it can generate waveforms. On power up, the waveform memory has no specific data. The sine waveform, being the default waveform on power on, is computed and loaded to the waveform memory as part of the reset procedure. From this moment on, every time that another standard waveform is selected, it is being computed and loaded to the waveform memory.

Waveforms are written from the same start address. Therefore, every time that a new waveform is selected, there is some minimal time for the processor to compute and download the data to the memory.

The PXA125 can be programmed to output one of nine standard waveform shapes: sine, triangle, square, pulse/ramp, sine(x)/x pulse, gaussian pulse, rising/decaying exponential pulse, noise and dc. There are some parameters associated with each waveform, which modify the shape of the waveform to better suit your needs. For example, different start phase for the sine waveform can be programmed for each channel to create phase offsets between the two instruments.

Arbitrary (User) Waveforms

The arbitrary waveform memory is capable of storing one or more user-defined waveforms. As was discussed before, the PXA125 is supplied with 2 Meg memory bank. There are up to 2 Meg points that can be allocated to one single waveform. On the other hand, there is no need to use the entire memory for only one waveform; The memory can be divided into smaller segments and loaded with different waveforms and the instrument can be programmed to output one segment at a time.

Loading data to arbitrary waveform memory can be a time-consuming task, especially if all 2 Meg points are loaded in one shot. The PXA125 utilizes a DMA (direct memory access) concept that speeds data transfer from host computer to the instrument.

Sequenced Waveforms

The sequence generator is a powerful tool that lets you link and loop segments in any way you desire. As a simple example of a sequenced waveform, look at Figures 1-5a through 1-5c. The waveforms shown in these figures were placed in memory segments 1, 2 and 3, respectively. The sequence generator takes these three waveforms links and loops them in a predefined order to generate the waveform shown in Figure 1-5d.

The sequence circuit is useful for generating long waveforms with repeated sections. The repeated waveform has to be programmed once and the repeater loops on this segment as many times as selected. When in sequenced mode, there is no loss of time between linked or looped segments.

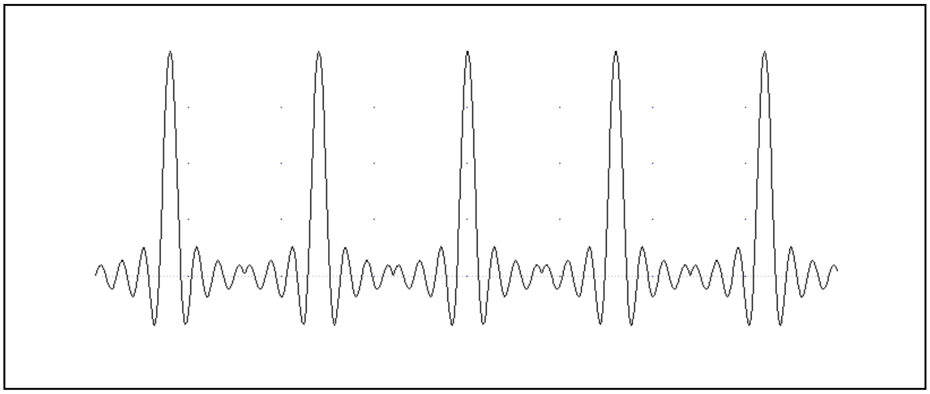


Figure 1-5a, Segment 1 – Sin (x)/x Waveform

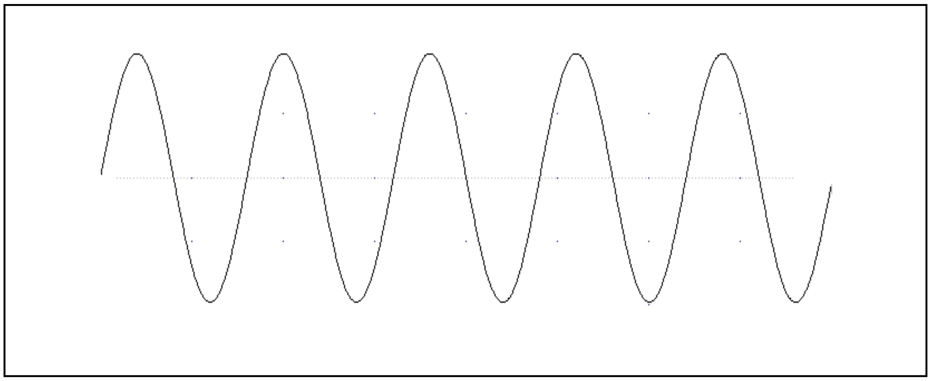


Figure 1-5b. Segment 2 – Sine Waveform

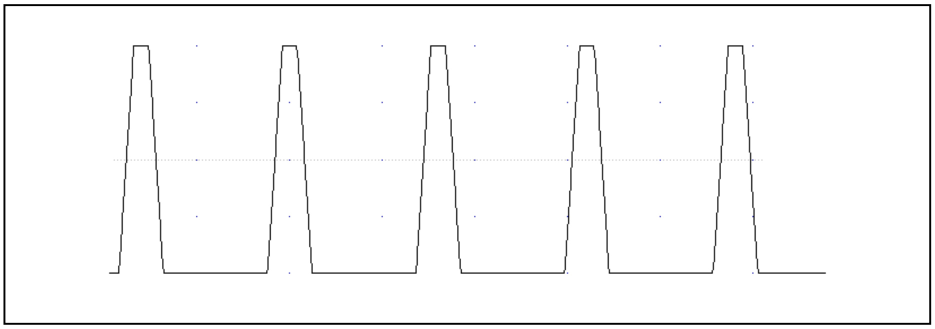


Figure 1-5c Segment 3 – Pulse Waveform

The following sequence was made of segment 2 repeated twice, segment 1 repeated four times, and segment 3 repeated two times.

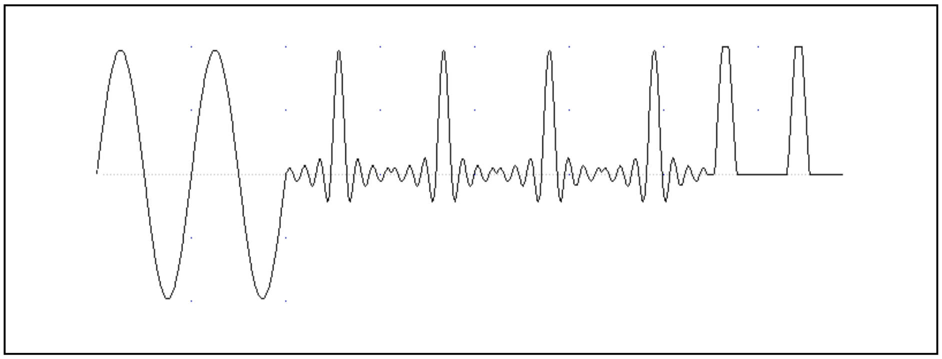


Figure 1- 5d. Sequenced Waveforms

Output State

The main outputs can be turned on or off. The internal circuit is disconnected from the output connector by a mechanical switch (relay). This feature is useful for connecting the main outputs to an analog bus. For safety reasons, when power is first applied to the chassis, the main output is always off.

Filters

Two filters are built into the PXA125. These filters are available for use in various applications such as the creation of high frequency sine waves and removing the staircase effect from waveforms that are generated with high frequency clock rates. The filters are also used for reconstructing the standard sine waveform.

Programming The PXA125

The PXA125 does not have front panel control capability. Also, waveform data and sequence tables must be loaded to the PXA125 from a host computer before it can be output arbitrary or sequenced waveforms. There are a number of ways to “talk” to the instrument. They all require that an appropriate software driver be installed in the host computer. The rest is a matter of practice and knowledge of the language in use. These topics are discussed in later chapters.

Low level programming of the PXA125 is accomplished using SCPI (Standard Commands for Programmable Instruments) language. Programming aspects are covered in Chapter 4.

Supplied with the PXA125 is a PC software package called ArbConnection. This software provides a user interface that allows interacting with and controlling the PXA125 directly. Details on how to use ArbConnection are given in Chapter 3.

Chapter 2
INSTALLATION

Installation Overview

This chapter contains information and instructions necessary to prepare the Model PXA125 for operation. Details are provided for initial inspection, grounding requirements, repackaging instructions for storage or shipment and installation information.

Unpacking and Initial Inspection

Unpacking and handling of the generator requires normal precautions and procedures applicable to handling of sensitive electronic equipment. The contents of all shipping containers should be checked for included accessories and certified against the packing slip to determine that the shipment is complete.

Safety Precautions

The following safety precautions should be observed before using this product and associated computer. Although some instruments and accessories would normally be used with non-hazardous voltages, there are situations where hazardous conditions may be present.

This product is intended for use by qualified persons who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. The following sections contain information and cautions that must be observed to keep the PXA125 operating in a correct and safe condition.

⚠CAUTION

For maximum safety, do not touch the product, test cables, or any other instrument parts while power is applied to the circuit under test. ALWAYS remove power from the entire test system before connecting cables or jumpers, installing or removing cards from the chassis. Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always keep your hands dry while handling the instrument.

Operating Environment

The PXA125 is intended for operation within a PXI chassis as a plug-in module. Ensure the PXI chassis being used to host the PXA 125 fully conforms to the latest PXI specifications.

The PXA125 is intended for indoor use and should be operated in a clean, dry environment with an ambient temperature within the range of 0 °C to 40 °C.

⚠ ***WARNING***

The PXA125 must not be operated in explosive, dusty, or wet atmospheres. Avoid installation of the module close to strong magnetic fields.

The design of the PXA125 has been verified to conform to EN 61010-1 safety standard per the following limits:
Installation (Overvoltage) Category I (Measuring terminals)
Pollution Degree 2

Installation (Overvoltage) Category I refers to signal level, which is applicable for equipment measuring terminals that are connected to source circuits in which measures are taken to limit transient voltages to an appropriately low level.
Pollution Degree 2 refers to an operating environment where normally only dry non-conductive pollution occurs. Occasionally a temporary conductivity caused by condensation must be expected

Power Requirements

The PXA125 operates from within a PXI chassis. DC Voltages are supplied to the instrument from the PXI backplane. The instrument requires a variety of DC voltages as outlined in the Specifications section (Appendix A). Ensure the PXI chassis is capable of delivering required voltages and has sufficient current to drive the generator.

⚠ ***CAUTION***

Disconnect power to the PXI Chassis before installing or removing the PXA125.

Grounding Requirements

To conform to the applicable safety and EMC requirements, ensure that the PXA125 instrument panel and the PXI chassis are “earth” grounded.

⚠ ***CAUTION***

The outer shells of the front panel terminals (OUTPUT, SYNC OUT, TRIG IN, SINEOUT, 10M REF IN) are connected to the instrument's chassis and therefore to the safety ground.

⚠ ***CAUTION***

Do not attempt to float the OUTPUT from ground as it may damage the PXA125 and other equipment connected to the PXA125 I/O connectors.

Calibration

The recommended calibration interval is one year. Calibration should be performed by qualified personnel only.

Abnormal Conditions

Operate the PXA125 only as intended by the manufacturer. If you suspect the PXA125 has been impaired, remove it from the PXI Chassis and secure against any unintended operation. The PXA125 protection is likely to be impaired if, for example, the instrument fails to perform the intended measurements or shows visible damage.

⚠ ***WARNING***

Any use of the PXA125 in a manner not specified by the manufacturer may impair the protection provided by the instrument

Long Term Storage or Repackaging For Shipment

If the instrument is to be stored for a long period of time or shipped immediately, proceed as directed below. If you have any questions, contact your local LeCroy representative or the LeCroy Customer Service Department.

1. Repack the instrument using the wrappings, packing material and accessories originally shipped with the unit. If the original container is not available, purchase replacement materials.
2. Be sure the carton is well sealed with strong tape or metal straps.
3. Mark the carton with the model and serial number. If it is to be shipped, show sending and return address on two sides of the box.

NOTE

If the instrument is to be shipped to LeCroy for calibration or repair, attach a tag to the instrument identifying the owner. Note the problem, symptoms, and service or repair desired. Record

the model and serial number of the instrument. Show the returned authorization order number (RMA) as well as the date and method of shipment. ALWAYS OBTAIN A RETURN AUTHORIZATION NUMBER FROM THE FACTORY BEFORE SHIPPING THE INSTRUMENT TO LeCroy.

Preparation For Use

Preparation for use include removing the instrument from the bag, installing the PXA125 inside the PXI chassis, copying instrument drivers to the computer and installing the graphical interface (ArbConnection).

Removing the Instrument from the Bag

The PXA125 is supplied in an antistatic bag. Check the seal on the bag to make sure the bag was not opened in a static-unsafe environment. Place the enveloped card on static free surface and hook yourself up with a grounding strap. Only then break the seal and remove the card from the envelope. Hold the card at the metal panel end. Refrain from touching the instrument with your finger at all times.

Installation

Plug the PXA125 into your PXI chassis and lift the extractor to the upright position. Push the card firmly until the metal panel makes contact with the metal edge of the PXI chassis. Using a suitable screwdriver, tighten the two retaining screws, top and bottom.

⚠ ***CAUTION***

Disconnect power to the PXI Chassis before installing or removing the PXA125. An attempt to insert or remove the instrument while the power is connected to the chassis will result in severe damage to the instrument and will automatically revoke your warranty.

⚠ ***CAUTION***

Once the PXA125 is installed in the chassis cover all remaining open slots to ensure proper airflow. Using the PXA125 without proper airflow will result in damage to the instrument.

Installing Instrument Drivers

The PXA125 is a Plug & Play instrument, meaning that after you install it in your PXI chassis, Windows will automatically detect its presence and will ask you to supply the appropriate drivers to operate this instrument. After you copy the drivers, Windows will add the drivers to the device manager and will assigns resources for the instrument.

The PXA125 is supplied with a CD that contains the *inf file for installing the necessary drivers for operating the instrument on your computer. Follow the instructions below to install the driver to your computer.

Minimum System Requirements

Minimum host system requirements for the instrument are as follows:

1. Windows 95/98/ME/NT/2000/XP
2. 32MB of RAM, 128MB or more recommended
3. 20MB hard drive space

Windows Software Installation

Note

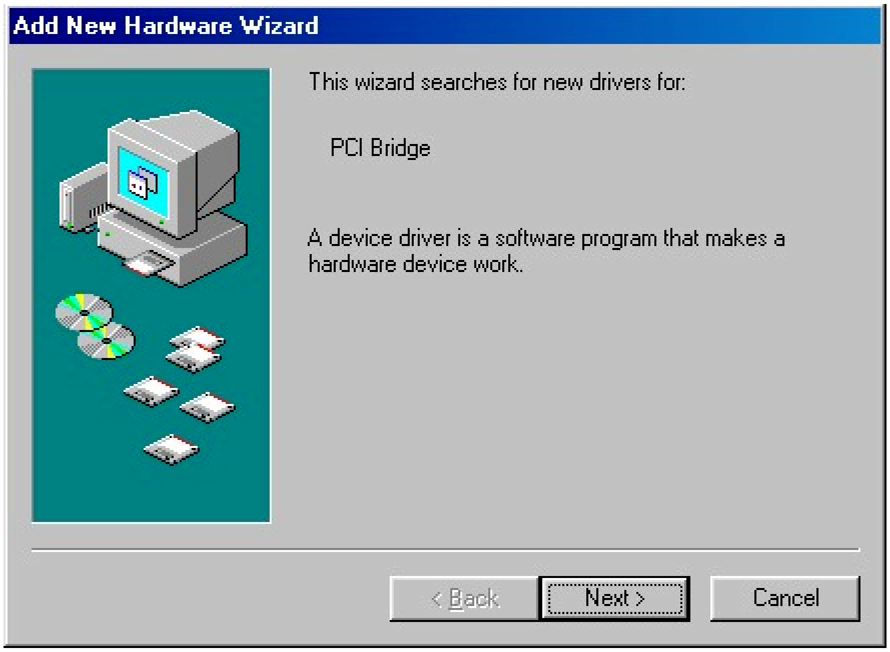
The install program will automatically detect older software version and you will be provided with a choice of either removing it and installing newer versions, or keeping your old installation intact. The install program will not remove instrument drivers. In case you want to remove the instrument driver from your computer, go to the Windows Device Manager (as shown in Figure 2-8) and remove the Arbitrary Waveform Generator Model PXA125 under the LeCroy heading.

In general, installation is very similar for different Windows variants however, you should follow the exact procedure as applicable for you're your computer. The following paragraphs will guide you through device driver installation for Windows 95/98, ME, Windows 2000/XP and Windows NT.

Windows 95/98/ME Device Driver Installation

A device driver is necessary for the PXA125 software to communicate with the PXI boards. Windows applications cannot communicate with PXI devices without a device driver installed. The PXA125's installation CD includes instrument drivers for supporting the required Windows platform. In Windows 95/98/ME, the installation package cannot automatically assign device drivers for PXI devices and therefore leaving for the Plug & Play Manager the responsibility for detecting devices and prompting the user for the correct driver. To assign a driver to a device, Windows refers to an *.INF file. The *.INF file provides instructions for Windows as to which driver files to install and which registry entries to insert. To install a driver for your PXA125, complete the following steps:

1. Power down your computer and PXI chassis.
2. Insert the PXI PXA125 board into a free PXI slot.
3. Power on your PXI chassis
4. Power on the computer. Windows should first detect the new hardware device with a “New Hardware Found” message box.
5. Windows displays the “Add New Hardware Wizard” as shown in Figure 2-1, which will search for new drivers
6. Follow the procedure as shown below.
7. After installing the PXI PXA125 successfully (see the previous section), you’ll be prompted to reboot your computer.



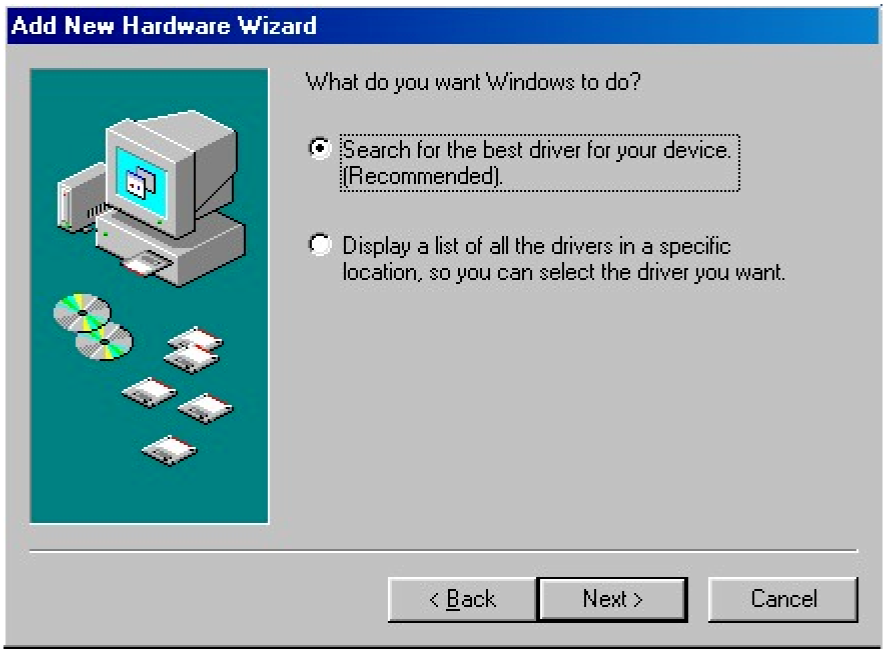
This wizard searches for new drivers for:

PCI Bridge

A device driver is a software program that makes a hardware device work.

Figure 2-1 – Add New Hardware Wizard

Press Next and select one of the options in the dialog box below. We recommend that you check the “Search for the best driver for your device” option as shown in Figure 2-2.



Add New Hardware Wizard

What do you want Windows to do?

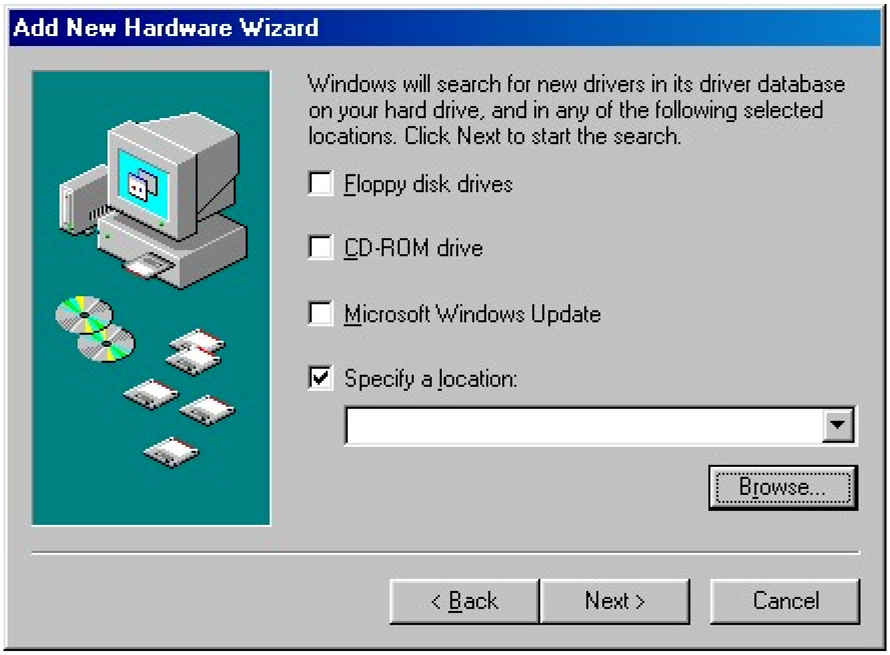
● Search for the best driver for your device. (Recommended).

○ Display a list of all the drivers in a specific location, so you can select the driver you want.

Figure 2-2 – Search for PXA125 Driver

Press Next and you'll now be prompted to select the location of the required *.INF file on your computer as shown in Figure 2-3.

The PXA125 is supplied with installation CD. Insert this CD into your CD-ROM drive and check the "Specify a location" option as shown below. If you already have the *.INF installed somewhere on your computer, you can specify your custom location. Either type in the complete path or click on the Browse button to find the path to your previous driver location.



Add New Hardware Wizard

Windows will search for new drivers in its driver database on your hard drive, and in any of the following selected locations. Click Next to start the search.

□ Floppy disk drives

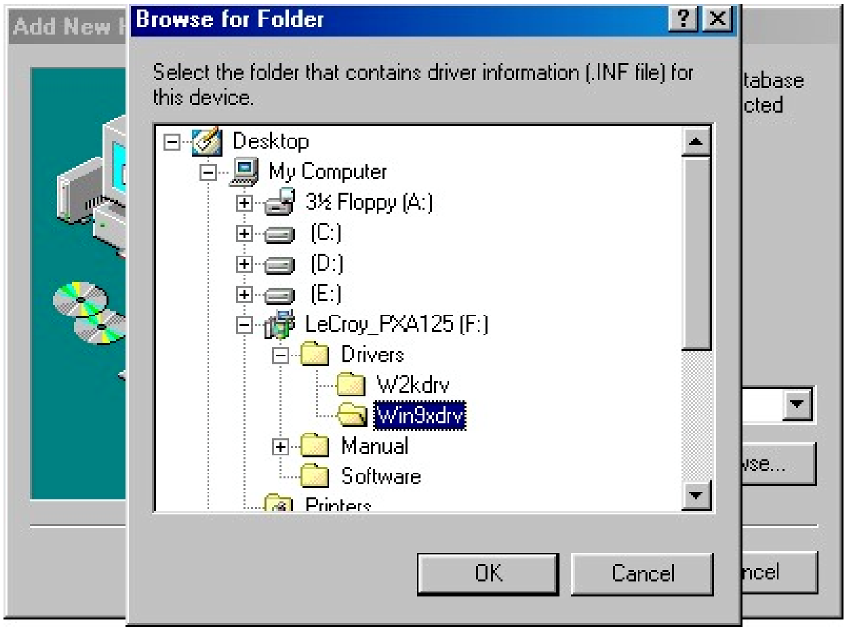
□ CD-ROM drive

□ Microsoft Windows Update

☑ Specify a location:

Figure 2-3– Specify PXA125 Driver Location

Assuming that this is your first-time installation, press Browse find the Win9xdrv subfolder that is located in the Drivers folder on the supplied CD, as shown in Figure 2-4.

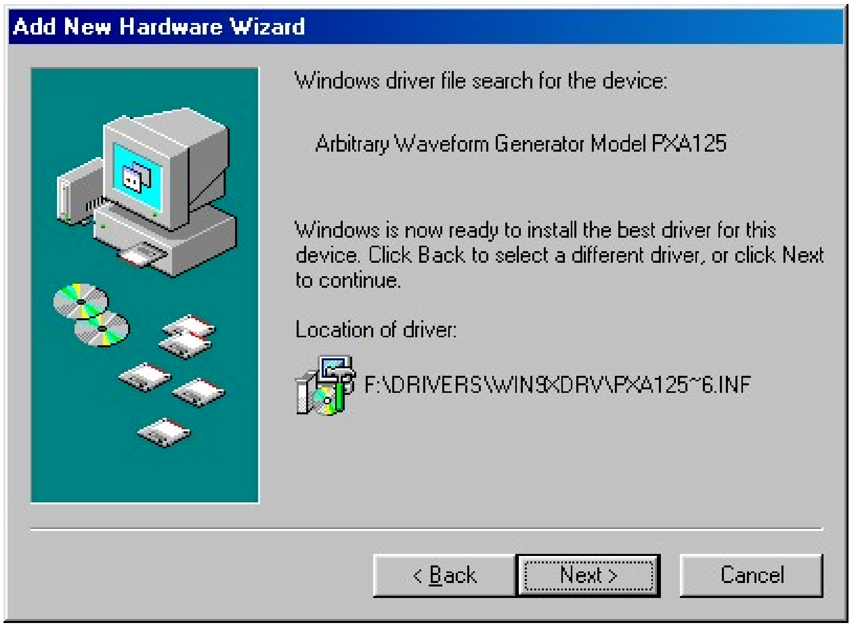


Browse for Folder

Select the folder that contains driver information (.INF file) for this device.

Figure 2-4 – Specify Subfolder for Your instrument

Press OK, then next; Windows will automatically detect the device driver and will show the next dialog box, as shown in Figure 2-5. Check the device description to make sure it agrees with your instrument description.



Add New Hardware Wizard

Windows driver file search for the device:

Arbitrary Waveform Generator Model PXA125

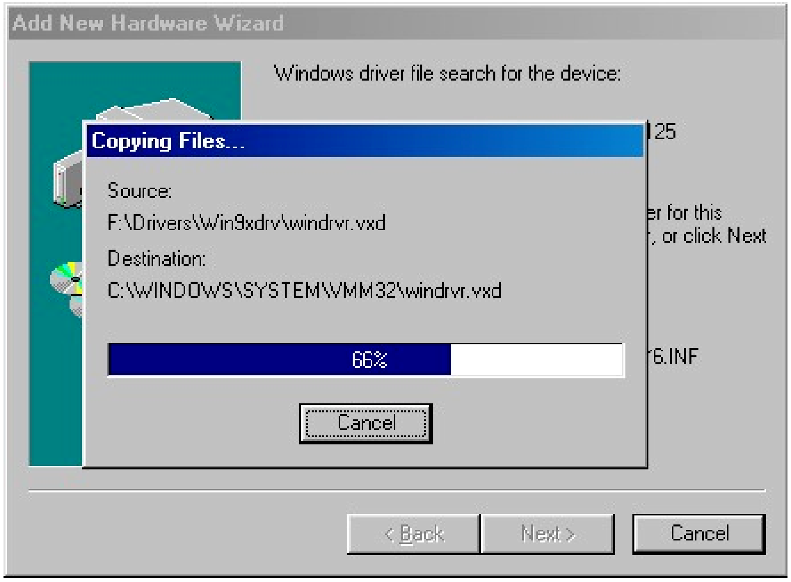
Windows is now ready to install the best driver for this device. Click Back to select a different driver, or click Next to continue.

Location of driver:

F:\DRIVERS\WIN9XDRV\PXA125~6.INF

Figure 2-5 – Device Driver Detected

Press Next and observe that Window copies the files from the CD-ROM drive to your hard drive, as shown in Figure 2-6.



Add New Hardware Wizard

Windows driver file search for the device:

Copying Files...

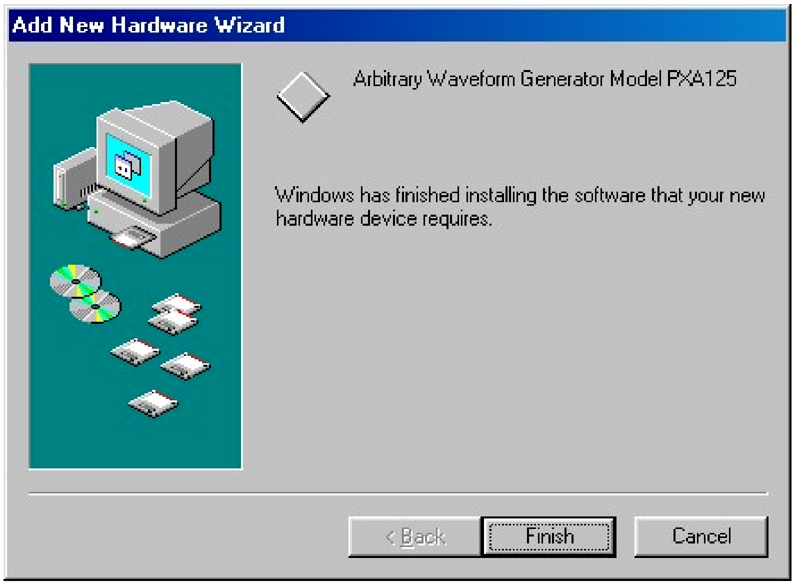
Source:
F:\Drivers\Win9xdrv\windrvr.vxd

Destination:
C:\WINDOWS\SYSTEM\VMM32\windrvr.vxd

66%

Figure 2-6. Copying Device Driver

After Windows has completed copying the device drivers, you can click on the Finish button to complete the installation process.



Add New Hardware Wizard

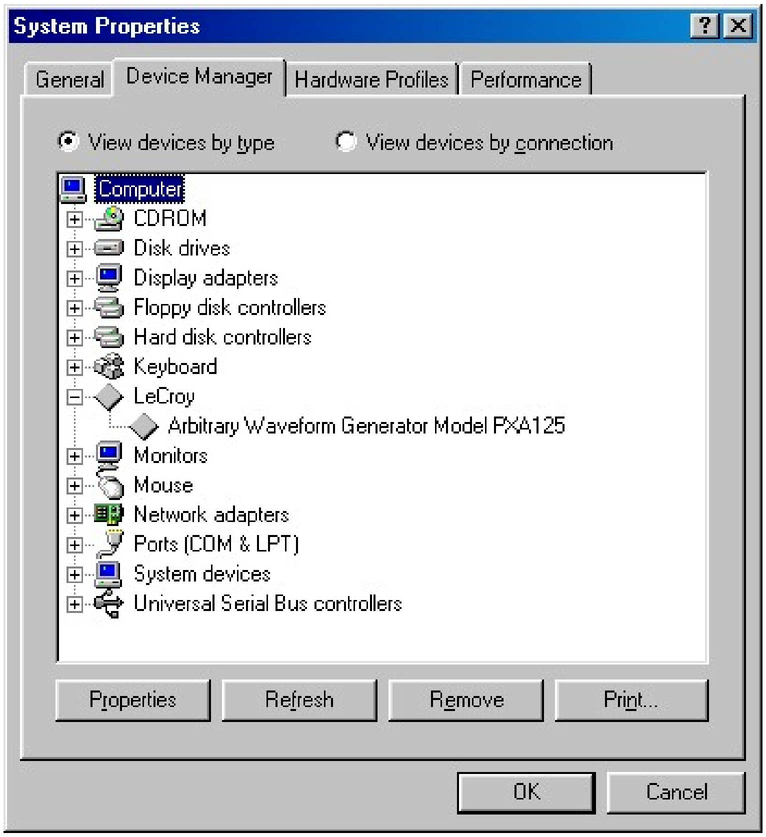
Arbitrary Waveform Generator Model PXA125

Windows has finished installing the software that your new hardware device requires.

Figure 2-7 – Finish Copying Device Driver

The next step is necessary to verify that the device driver was installed properly and is displayed correctly in the System Device Manger. Do the following:

1. Click on the Start button, then Settings and then Control Panel.
2. In the Control Panel dialog box, find the icon labeled System.
3. Double-click on the System icon and then on the Device Manager tab.
4. Click on LeCroy. If you installed the driver properly, your device manager should show the Arbitrary Waveform Generator Model PXA125 as shown in the figure 2-8 below.



● View devices by type ○ View devices by connection

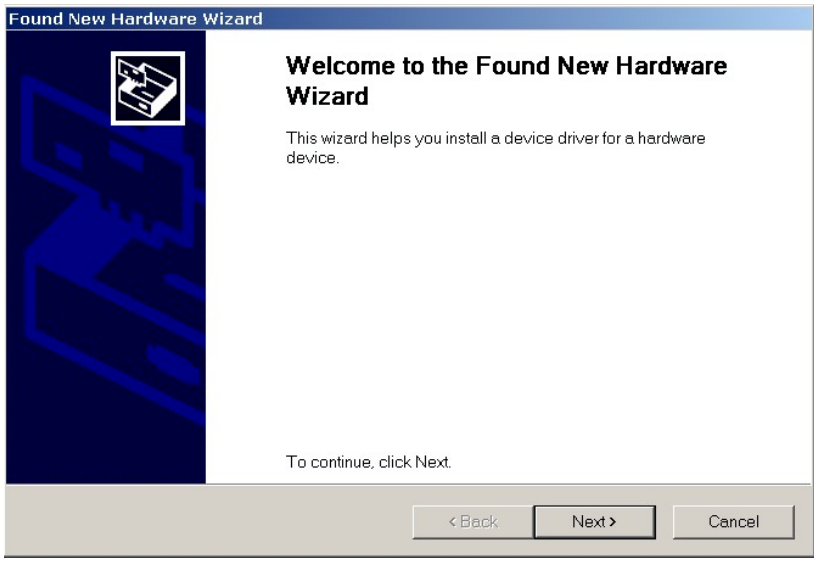
Figure 2-8. Checking Driver Installation

Click on “Cancel” to remove the System Properties dialog box from the screen.

Windows 2000/XP Device Driver Installation

A device driver is necessary for the PXA125 software to communicate to the PXI boards. Windows applications cannot communicate to PXI devices without a device driver installed. The PXI PXA125's installation CD includes instrument drivers for supporting the required Windows platform. In Windows 2000/XP, the installation package cannot automatically assign device drivers for PXI devices and therefore leaving for the Plug & Play Manager the responsibility for detecting devices and prompting the user for the correct driver. To assign a driver to a device, Windows refers to an *.INF file. The *.INF file provides instructions for Windows as to which driver files to install and which registry entries to insert. To install a driver for your PXA125, complete the following steps:

1. Power down your computer and PXI chassis.
2. Insert the PXI PXA125 board into a free PXI slot.
3. Power on your PXI chassis
4. Power on the computer. Windows should first detect the new hardware device with a "Found New Hardware Wizard" message box.
5. Windows then displays the "Found New Hardware Wizard" as shown in Figure 2-9, which will search for new drivers
6. Follow the procedure as shown below.



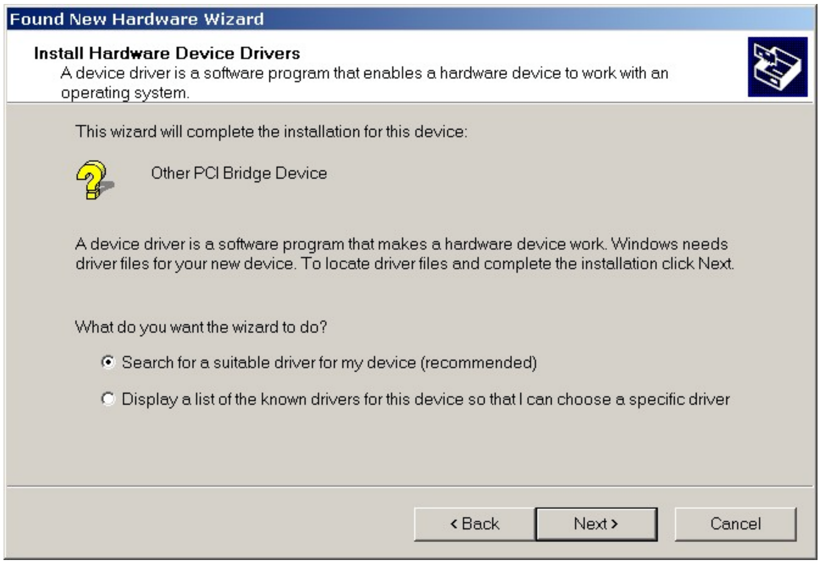
Welcome to the Found New Hardware Wizard

This wizard helps you install a device driver for a hardware device.

To continue, click Next.

Figure 2-9 – The Welcome to the Found New Hardware Wizard

Press Next and select one of the options in the dialog box below. We recommend that you check the "Search for a suitable driver for my device" option as shown in Figure 2-10.



Install Hardware Device Drivers
A device driver is a software program that enables a hardware device to work with an operating system.

This wizard will complete the installation for this device:

Other PCI Bridge Device

A device driver is a software program that makes a hardware device work. Windows needs driver files for your new device. To locate driver files and complete the installation click Next.

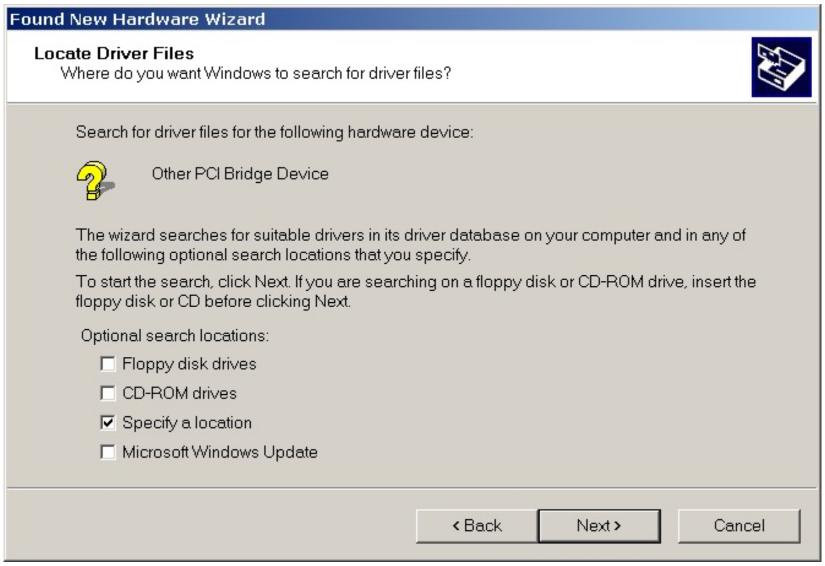
What do you want the wizard to do?

● Search for a suitable driver for my device (recommended)

○ Display a list of the known drivers for this device so that I can choose a specific driver

Figure 2-10 – Install Hardware Device Drivers

Press Next and you'll now be prompted to select the location of your driver files on your computer as shown in Figure 2-11. Check the "Specify a Location" option only.



Locate Driver Files
Where do you want Windows to search for driver files?

Search for driver files for the following hardware device:

Other PCI Bridge Device

The wizard searches for suitable drivers in its driver database on your computer and in any of the following optional search locations that you specify.

To start the search, click Next. If you are searching on a floppy disk or CD-ROM drive, insert the floppy disk or CD before clicking Next.

Optional search locations:

□ Floppy disk drives
□ CD-ROM drives
☑ Specify a location
□ Microsoft Windows Update

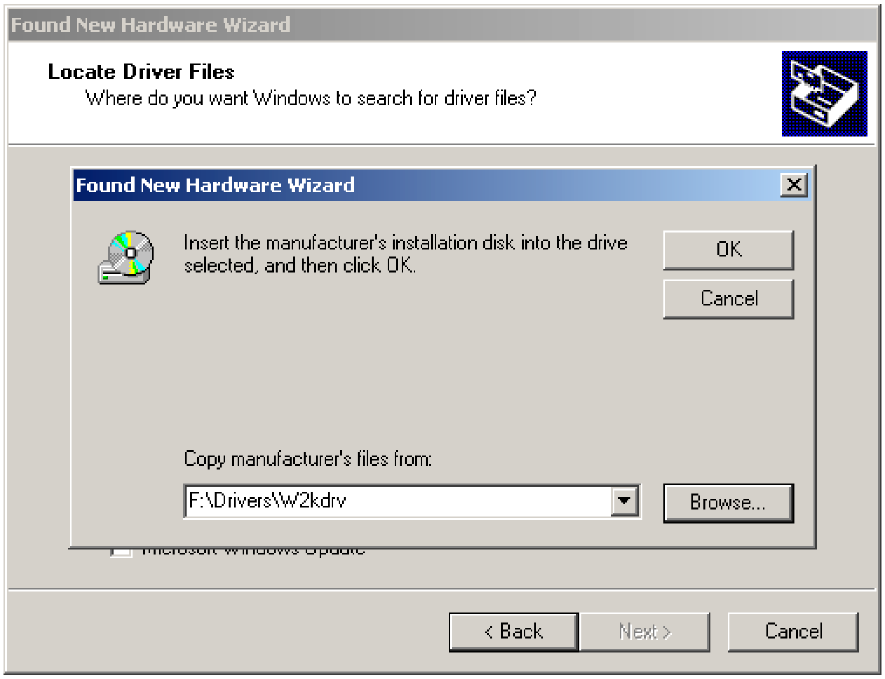
Figure 2-11 – Locate Driver Files

The PXA125 is supplied with an installation At this time, CD. Insert this CD into your CD-ROM drive. If you already have the instrument drivers installed somewhere on your computer, you can specify your custom location. Either type in the complete path or click on the Browse button to identify your driver location.

NOTE

After you insert the installation CD Windows 2000/XP will automatically load the autorun program and the CD interface will show on your monitor, similar to what you see in Figure 2-16. The CD menu is not required for the driver installation so you can either leave the menu on screen and ignore it for now or click on Exit to remove the application from your desktop. The installation CD must be left in the CD-Rom drive for the continuation of the installation process.

Press Browse… and specify the subfolder W2kdrv located in the Drivers folder on your installation CD, as shown in Figure 2-12. Press OK to acknowledge the selected path.


Found New Hardware Wizard

Locate Driver Files

Where do you want Windows to search for driver files?

Found New Hardware Wizard

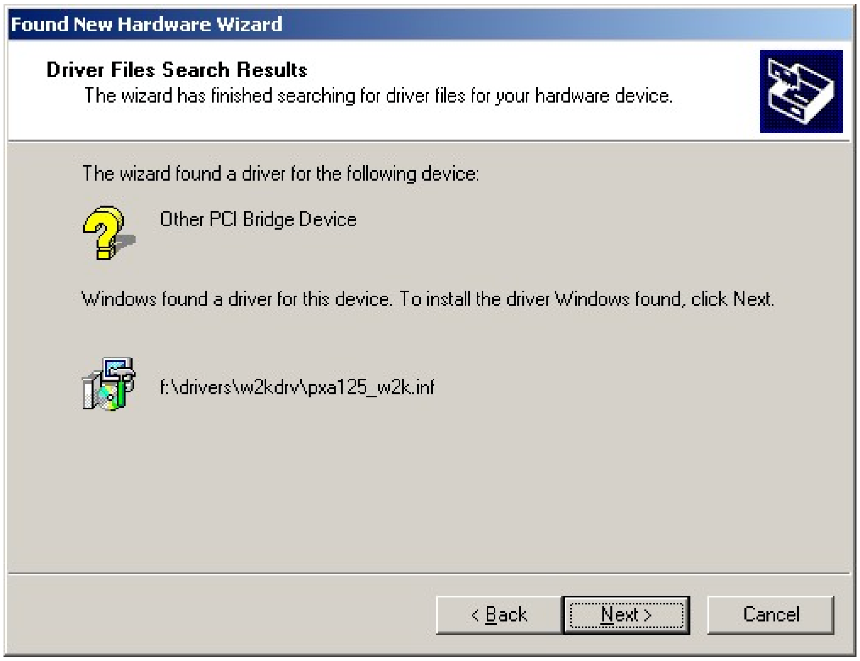
Insert the manufacturer's installation disk into the drive selected, and then click OK.

Copy manufacturer's files from:

F:\Drivers\W2kdrv

Figure 2-12 – Copying Device Drivers

Windows displays the Driver Files Search Results as shown in Figure 2-13. Click Next and then Finish to complete the installation process.



Found New Hardware Wizard

Driver Files Search Results

The wizard has finished searching for driver files for your hardware device.

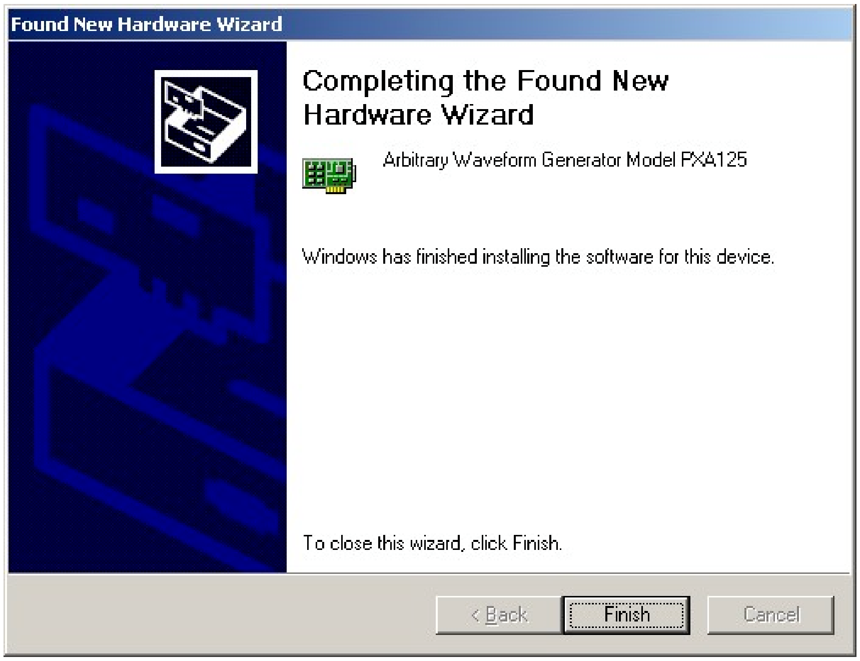
The wizard found a driver for the following device:

Other PCI Bridge Device

Windows found a driver for this device. To install the driver Windows found, click Next.

f:\drivers\w2kdrv\pxa125_w2k.inf

Figure 2-13 – Driver Files Search Results



Found New Hardware Wizard

Completing the Found New Hardware Wizard

Arbitrary Waveform Generator Model PXA125

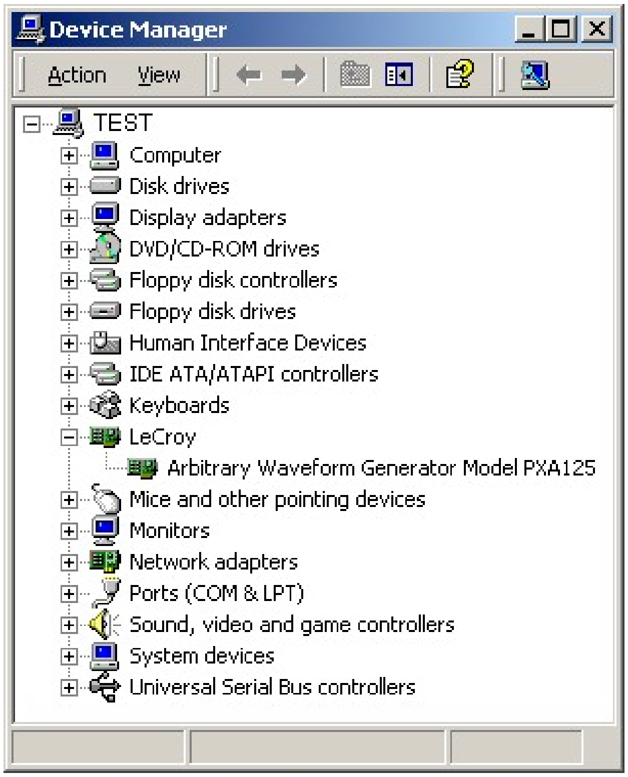
Windows has finished installing the software for this device.

To close this wizard, click Finish.

Figure 2-14 – Completing the Found New Hardware Wizard

The next step is necessary to verify that the device driver was installed properly and is displayed correctly in the System Device Manger. Do the following:

1. Click on the Start button, then Settings and then Control Panel.
2. In the Control Panel dialog box, find the icon labeled System.
3. Double click on the System icon and then on the Hardware
4. On the Hardware tab click on the Device Manager button.
5. Click on LeCroy. If you installed the driver properly, your device manager should show the Arbitrary Waveform Generator Model PXA125 as shown in figure 2-15.



- TEST
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Floppy disk controllers
  - Floppy disk drives
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Keyboards
  - LeCroy
    - Arbitrary Waveform Generator Model PXA125
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers

Figure 2-15 – Device Manager

Click on "X" to remove the System Properties dialog box from the screen.

Windows NT Device Driver Installation

A device driver is necessary for the PXA125 software to communicate with the PXI boards. Windows applications cannot communicate with PXI devices without a device driver installed. The PXA125's installation CD includes instrument drivers for supporting various Windows platforms.

Since Windows NT does not have Plug & Play capability, the installation process is completely different than how it's done on other Windows platforms. Actually, the Windows NT installation is transparent and automatic while you install some software utilities that are necessary to operate and detect the instrument on your PC. These utilities are called ArbConnection and ArbDetector. Detailed description of these utilities is given in other sections of the manual however, if you are an NT user, you must install these utilities before your NT system will be able to detect and control the PXA125.

To install your Windows NT driver and other software utilities, complete the following steps:

1. Power down your computer and PXI chassis.
2. Insert the PXA125 board into a free PXI slot.
3. Power on your PXI chassis
4. Power on your computer
5. Insert the CD into your CD-ROM drive.
6. Start installing the software using the procedure for "Installing ArbConnection and ArbDetector" utilities as described below.

Installing ArbConnection and ArbDetector

ArbConnection and ArbDetector are two software utilities that must be installed on your computer for you to be able to operate and control the PXA125. ArbConnection has three basic function: 1) Control instrument setups, 2) Create, edit and download complex arbitrary waveforms to your instrument and 3) Create, edit and download complex (frequency) modulating waveforms.

Advanced users may want to write their own applications to control the PXA125 functions and features. For this purpose, C++ libraries and DLL's are available on the CD and can be copied to your application folder. However, if you are first time user and can't wait to get your hands on some controls and waveforms, it is recommended that you install ArbConnection now. ArbConnection and waveform generation aspects are covered in other sections of this manual.

The ArbDetector program contains device initialization and interfacing links between your computer's board and the PXA125's PXI interface chip and therefore, ArbDetector must be installed and invoked whenever you want to use the PXA125, even if you are running your own application and regardless of which Windows platform is installed on your computer.

Windows NT users must install the ArbConnection and ArbDetector utilities as part of the basic installation process as described in the Windows NT Device Driver Installation section of this manual. The installation process of these two programs will automatically install device drivers on Windows NT systems.

To proceed with the installation process, insert the supplied CD into your CD-ROM drive and wait for a GUI (Graphical User Interface, as shown in Figure 2-16) to pop up on your screen. If you wait for a long time and the GUI fails to load it is possible that your computer is not set up to load CD programs automatically. In this case, Invoke Run and type:

```
N:\Autorun.exe  (where N is your CD drive letter)
```

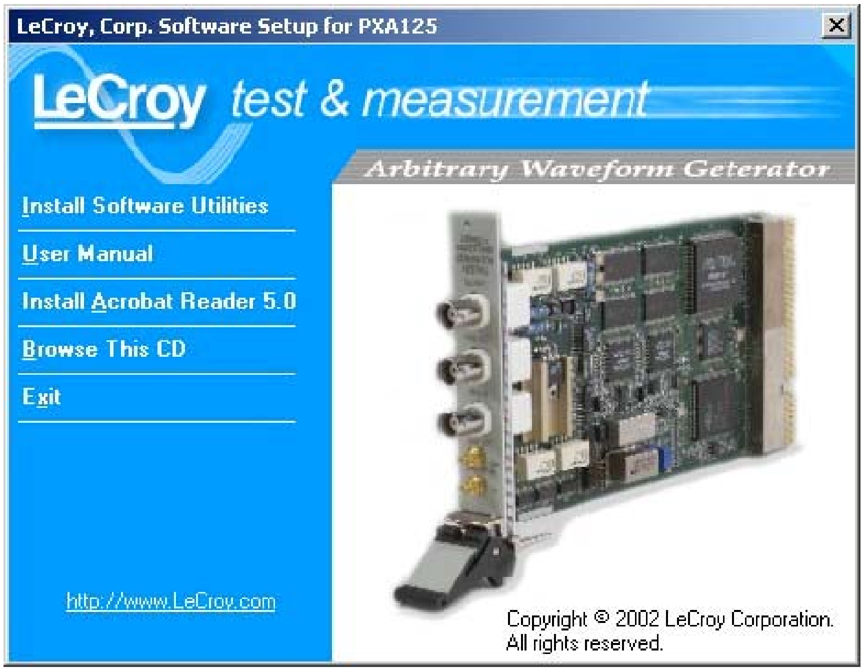
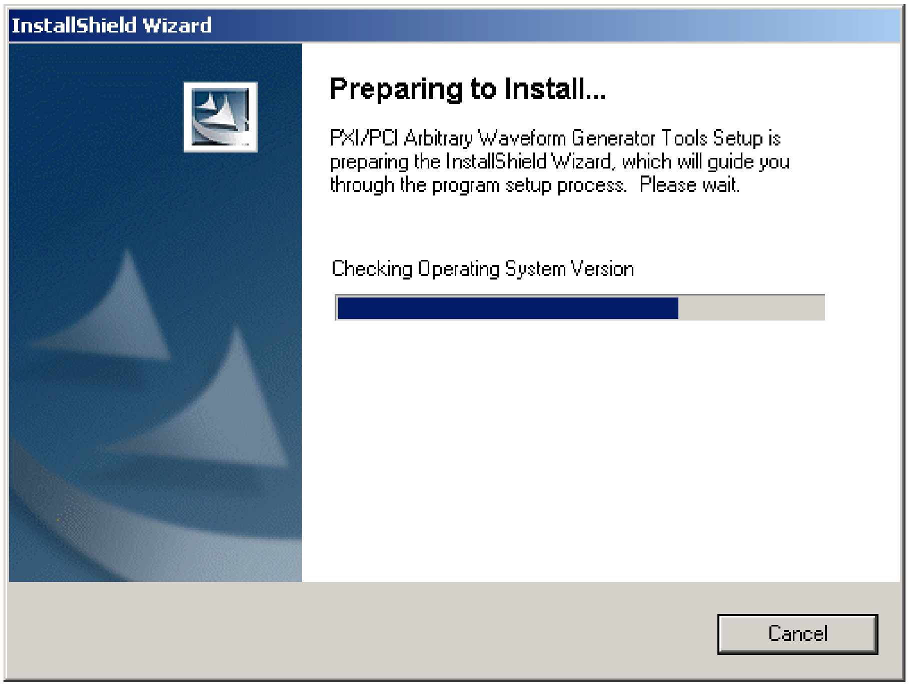

LeCroy, Corp. Software Setup for PXA125
LeCroy test & measurement
Arbitrary Waveform Geterator
Install Software Utilities
User Manual
Install Acrobat Reader 5.0
Browse This CD
Exit
http://www.LeCroy.com
Copyright © 2002 LeCroy Corporation.
All rights reserved.

Figure 2-16 – CD's GUI

To install the software click on "Install Software Utilities" and follow the install program as shown in Figure 2-17. The preparing phase will check few things such as if your computer has any previous installations installed on.

If you are an advanced user and in preparation to write your own code and application, you'll have to copy support libraries and some DLL's from the installation CD. In this case, select the "Copy Developer Libraries" option and specify the folder location for your application. The function of the libraries and other programming aspects are cover in the programming section of this manual.



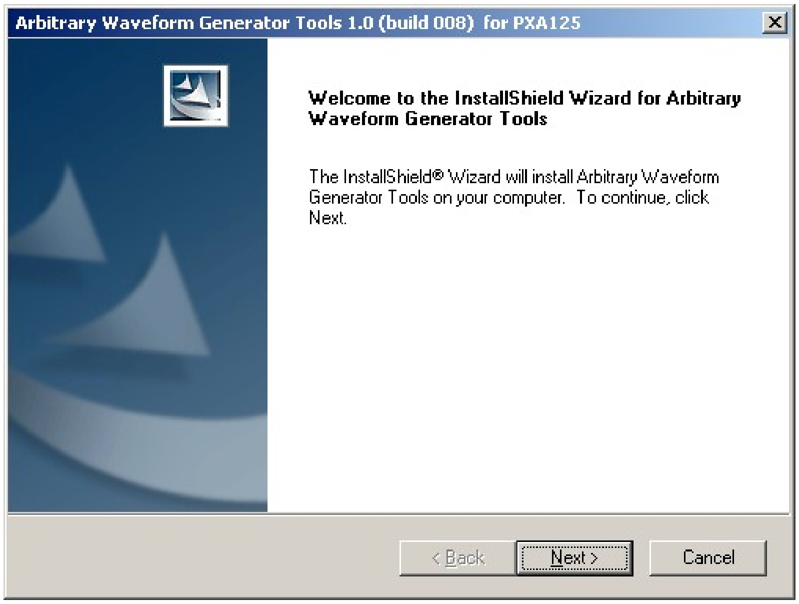
Preparing to Install...

PXI/PCI Arbitrary Waveform Generator Tools Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.

Checking Operating System Version

Figure 2-17 – Install preparation

After finishing the "Preparing to Install" phase, the install wizard shown in Figure 2-18 will take you to the first installation step assuming that no problems were detected.



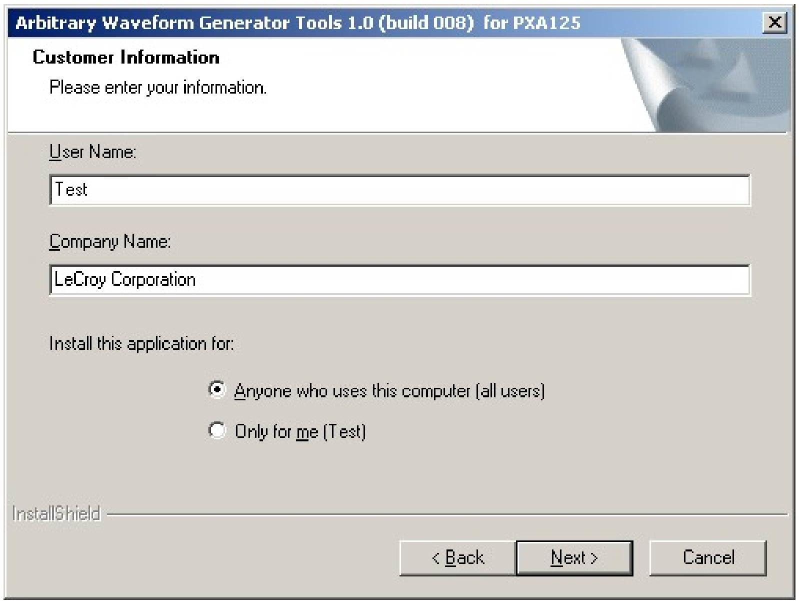
Arbitrary Waveform Generator Tools 1.0 (build 008) for PXA125

Welcome to the InstallShield Wizard for Arbitrary Waveform Generator Tools

The InstallShield® Wizard will install Arbitrary Waveform Generator Tools on your computer. To continue, click Next.

Figure 2-18 – First Installation Step

Press Next and type the customer details at the "Customer Information" Window, as shown in Figure 2-19.



Arbitrary Waveform Generator Tools 1.0 (build 008) for PXA125

Customer Information

Please enter your information.

User Name: Test

Company Name: LeCroy Corporation

Install this application for:

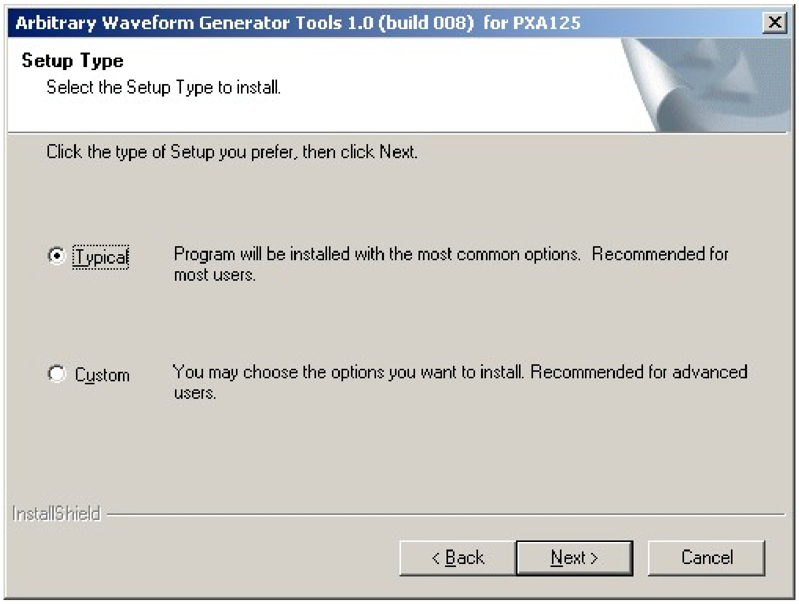
● Anyone who uses this computer (all users)

○ Only for me (Test)

Figure 2-19 – Customer Information Step

After typing the customer details press Next and select the Setup Type. You can select from three options: 1) Select PXA125 if you purchased and are installing the PXA125, or 2) Select Custom if you are an advanced user and want to refine your installation process.

We recommend that you install using options 1 or 2 above and complete the installation process by clicking on the Next button and then Finish. If you select the Custom option, then you'll have to type in some other parameters such as new path for your destination folder and choosing which feature to install, as shown in Figures 2-21 and 2-22.



Arbitrary Waveform Generator Tools 1.0 (build 008) for PXA125

Setup Type

Select the Setup Type to install.

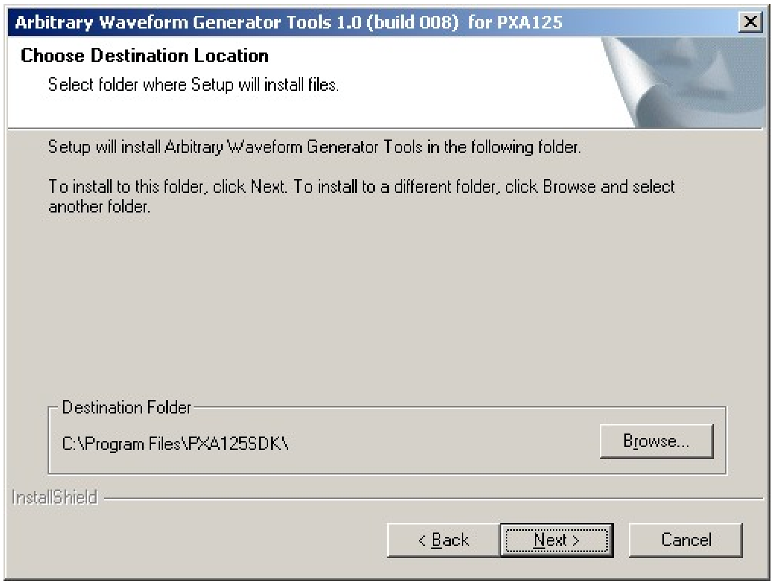
Click the type of Setup you prefer, then click Next.

● Typical — Program will be installed with the most common options. Recommended for most users.

○ Custom — You may choose the options you want to install. Recommended for advanced users.

Figure 2-20 – Selecting Setup Type

If you select the Custom option, proceed to change your Destination Folder, as shown in Figure 2-21, by pressing the Browse button, select the appropriate path, press OK and then press Next.


Arbitrary Waveform Generator Tools 1.0 (build 008) for PXA125

Choose Destination Location

Select folder where Setup will install files.

Setup will install Arbitrary Waveform Generator Tools in the following folder.

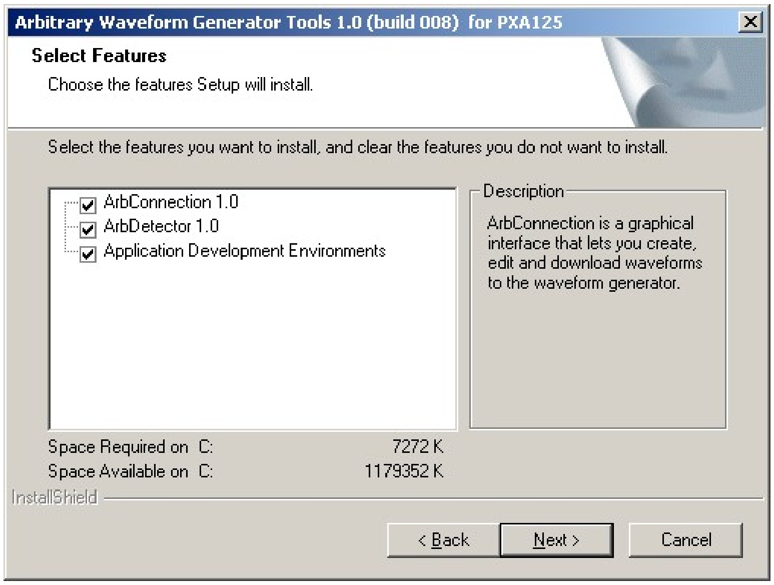
To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder

C:\Program Files\PXA125SDK\

Figure 2-21 – Selecting Destination

To select the Installed Features, as shown in Figure 2-22, select the appropriate feature by clicking on.


Arbitrary Waveform Generator Tools 1.0 (build 008) for PXA125

Select Features

Choose the features Setup will install.

Select the features you want to install, and clear the features you do not want to install.

☑ ArbConnection 1.0
☑ ArbDetector 1.0
☑ Application Development Environments

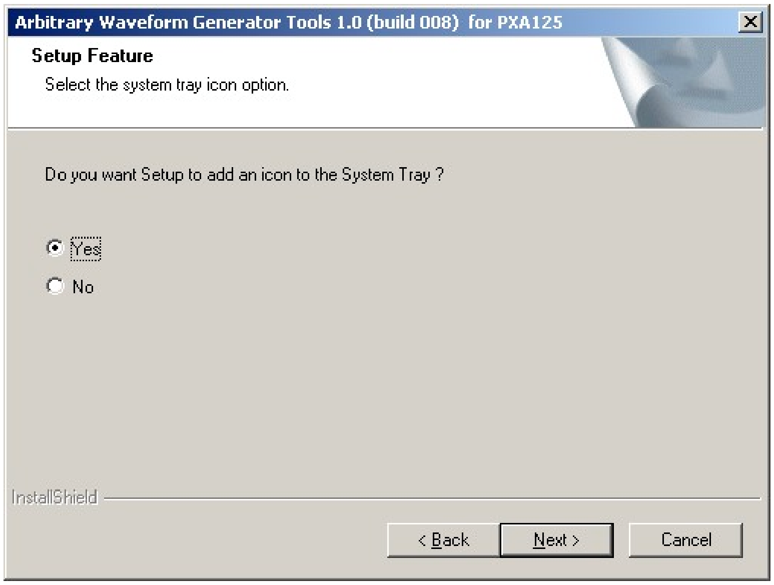
Description

ArbConnection is a graphical interface that lets you create, edit and download waveforms to the waveform generator.

Space Required on C: 7272 K
Space Available on C: 1179352 K

Figure 2-22 – Selecting Features

Just before you finish with the installation, you'll have to decide if you want an icon added to the system tray. The system tray is located on the right lower side of the screen that has icons that are loaded there automatically every time your computer is powered up. If you select the "No" option, ArbConnection will still launch at startup but the icon will not show and then, to access ArbDetector you'll have to invoke it from the Start → Programs → StartUp Menu. The ArbDetector icon selection dialog box is shown in Figure 2-23.


Arbitrary Waveform Generator Tools 1.0 (build 008) for PXA125

Setup Feature

Select the system tray icon option.

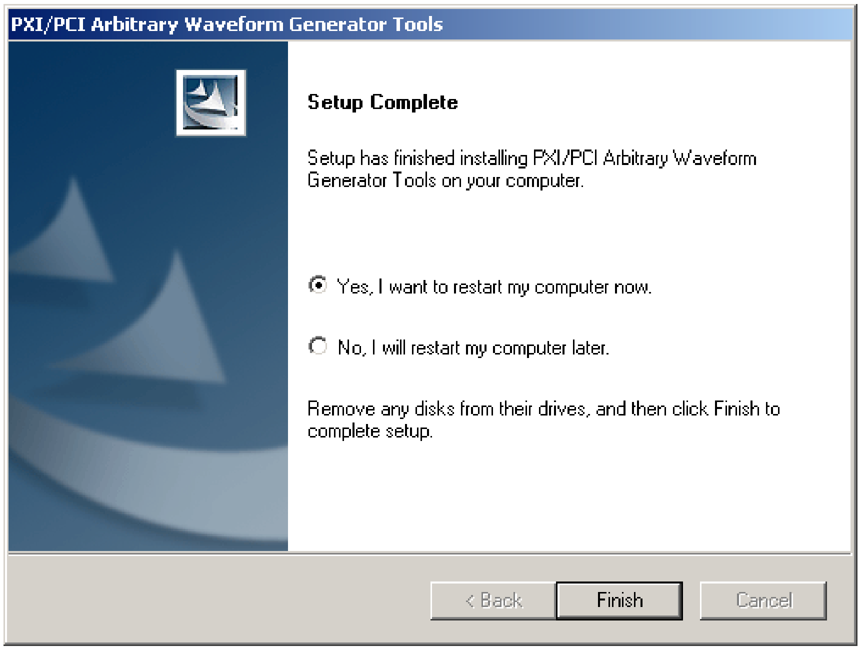
Do you want Setup to add an icon to the System Tray ?

● Yes

○ No

Figure 2-23 – Selecting ArbDetector Icon options

The final step to complete the installation process, you'll be prompted to restart your computer. You can select to either restart your computer immediately or do it later, but remember that the software will not function properly if you do not restart your computer.



PXI/PCI Arbitrary Waveform Generator Tools

Setup Complete

Setup has finished installing PXI/PCI Arbitrary Waveform Generator Tools on your computer.

● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

Figure 2-24 – Setup Complete

The ArbDetector

The ArbDetector detects all arbitrary waveform generators that are plugged into the PXI chassis and lets you review instrument identity, serial number, installed options etc. You may also use the device detector to fine-tune your controller type, chassis size and slot location.

Before you can start using the PXA125, you must make sure that the ArbDetector has been invoked. Look at your right lower end of your screen and identify the Startup bar. The ArbDetector icon looks as shown in Figure 2-25.

The ArbDetector is automatically invoked from the startup menu after you power on your computer. If, for any reason you cannot find the ArbDetector icon, you'll have to invoke it manually. From the Start button select Programs, then Startup and finally the ArbDetector command. The detector will be invoked and the icon placed at the startup tray.

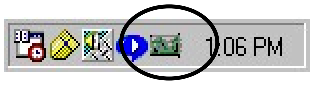


Figure 2-25. The ArbDetector Icon at the Startup Tray

When you click on the ArbDetector icon with your right mouse button, some options will show that will allow you to open the ArbDetector program, Reset all instruments to factory defaults, Delete al assigned slot locations and Verify users in case you want to Exit the program. The ArbDetector icon options is shown in Figure 2-26.

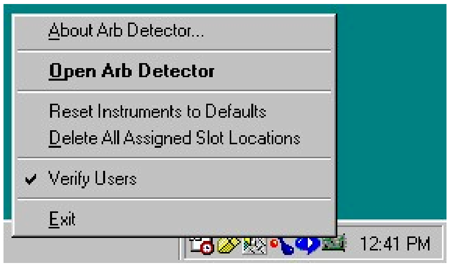
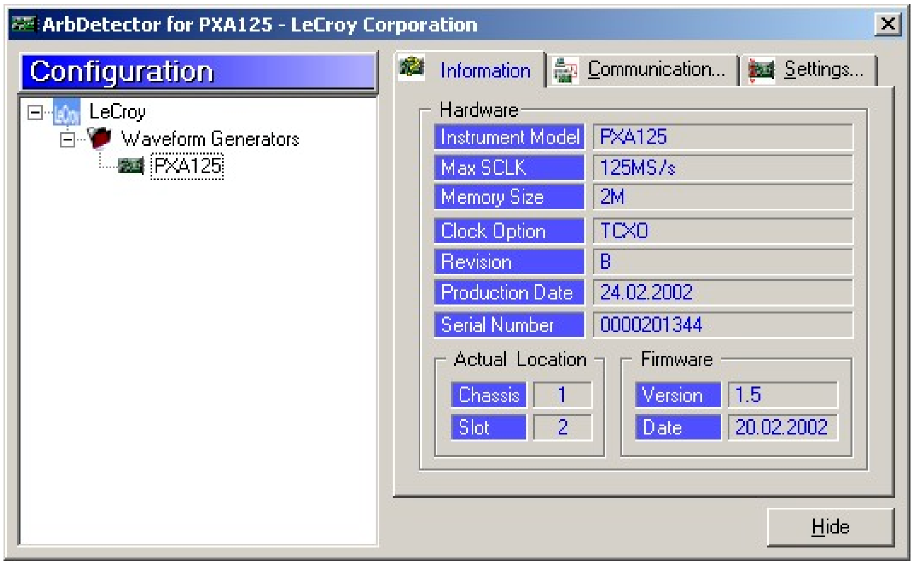


Figure 2-26. ArbDetector Icon Options

Using the ArbDetector

To invoke the ArbDetector program, either double-click on the icon or open the options window as shown in Figure 2-26 and click on the Open ArbDetector command. Doing so, the dialog box as shown in Figure 2-27 will open. If you cannot see the icon in your system tray, invoke the program from the StartUp menu.



Hardware

Instrument Model	PXA125
Max SCLK	125MS/s
Memory Size	2M
Clock Option	TCXO
Revision	B
Production Date	24.02.2002
Serial Number	0000201344

Actual Location

Chassis	1
Slot	2

Firmware

Version	1.5
Date	20.02.2002

Figure 2-27. The ArbDetector Dialog Box

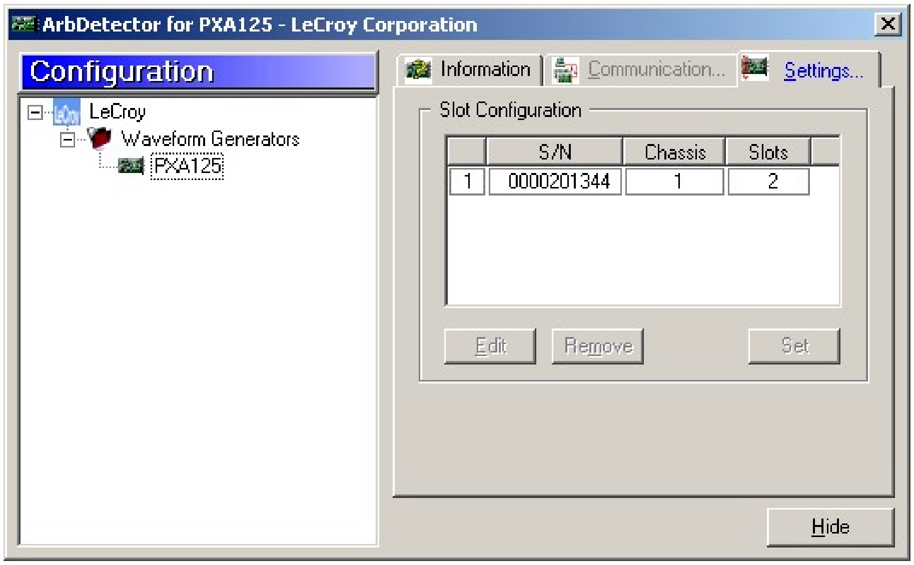
The ArbDetector dialog box is divided into two sections. The left side looks just like the familiar explorer program. The purpose of this explorer is to show how many devices are currently located inside your PXI system. This program will detect waveform generators only so please do not try using this program to detect other instruments.

There are three tabs on the right side of the dialog box. Information, Communication… and Settings… These tabs can only be used if at least one instrument has been found in the system. To be able to access these tabs, click on one of the instruments on the left side to highlight it, as shown in Figure 2-27. The information tab is automatically updated from the instrument flash memory with the following information: model number, maximum sample clock rate, installed memory option, installed clock option, hardware revision, production date and serial number. It also will show the firmware version and its release date.

Notice the Actual Location fields. There are two parameters there that are user controllable. These are the chassis number and actual slot location. These parameters can be programmed in the Setting… tab. If you have only one instrument in your system and do not care about its actual and physical location in the system, then you do not have to touch these fields and the instrument will perform flawlessly, even with these fields empty. In Figure 2-27, the instrument is located in chassis number 1 and slot number 2.

Configuring Actual Slot Location

The ArbDetector is flexible enough to allow you to specify the actual slot where your instrument resides. This is important in systems that have multiple chassis and multiple arbs, regardless if you are using chassis connected in “star” or serial format. After you specify the location of the instrument, external applications like ArbConnection can access these parameters and program just the instrument you want to access. To program slot location, click on the Settings… tab. The Settings tab is shown in Figure 2-28.

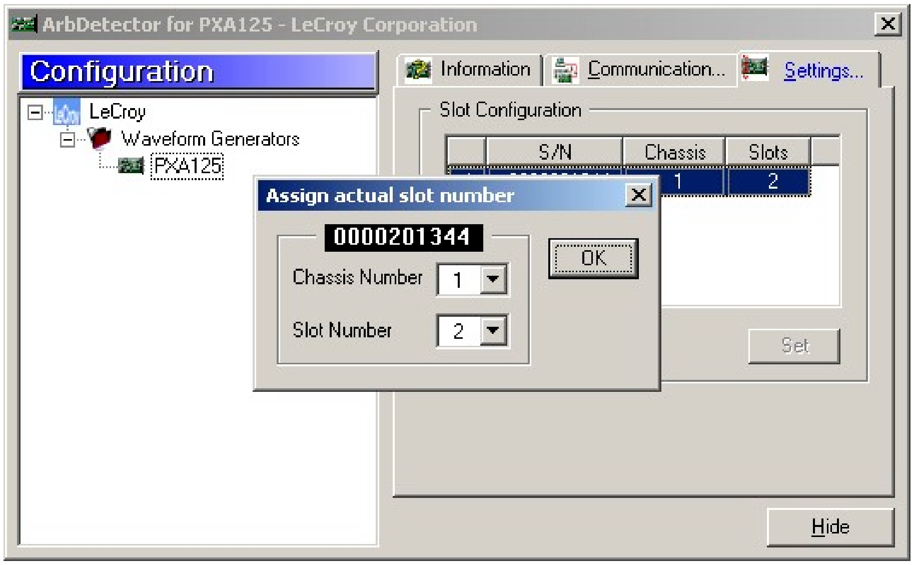


Slot Configuration

	S/N	Chassis	Slots
1	0000201344	1	2

Figure 2-28. The Settings Tab

The Slot Configuration as shown in Figure 2-28 is the end result after you assign the Chassis number and Slot parameters. To add or edit these parameters, click on the Edit button. The Edit dialog box will display as shown in Figure 2-29.



0000201344

Chassis Number: 1

Slot Number: 2

Figure 2-29. Editing Chassis Number and Slot location

In the Slot Number field select your actual location of your instrument and in the Chassis Number select the actual chassis where your instrument resides. To complete the chassis and slot assignment, click on OK. The dialog box will be removed from the screen the and Slot configuration parameters will be updated with the information you programmed just now. Notice the 10-digit number above the slot number. This is the serial number of your instruments as was detected by the ArbDetector. To make sure this is the right instrument you want to allocate, remove the instrument from your chassis and compare to the serial number as printed on the card.

Testing Communications With your Instrument

Final operation with your ArbDetector allows you to test communication with your instrument. If you click on the Communication… tab, the tab, as shown in Figure 2-30 will show. The Command Editor field lets you write SCPI commands that control instrument functions and parameters. SCPI programming is covered in Chapter 4 of this manual. Connect your instrument to an oscilloscope and try these simple commands just to see you're your instrument responds correctly:

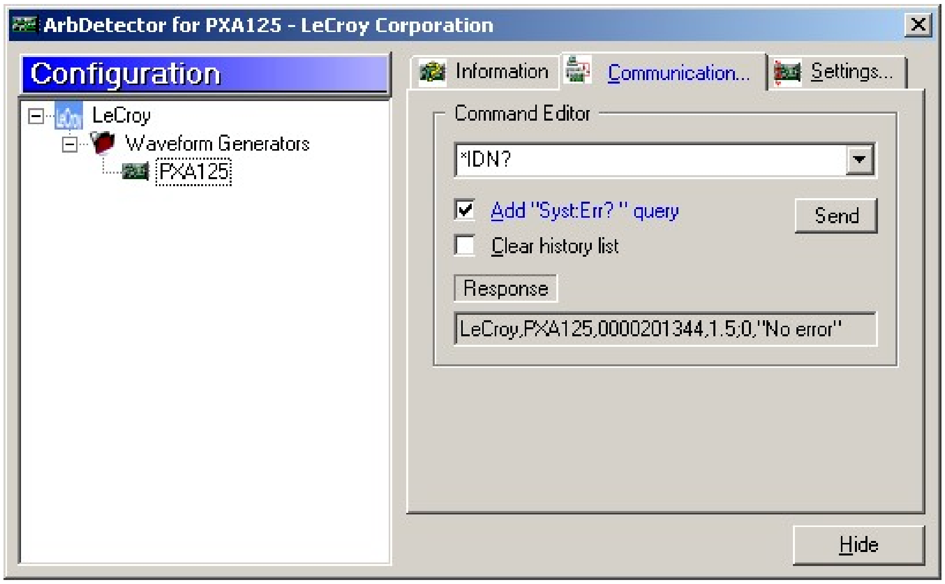
output on	will turn on your output and sinewave will be
frequency 2e6	will change output frequency to 2 MHz
volt 5	will change output amplitude to 5 Vp-p

If you are not sure about the commands and want the Command Editor to alert on errors, check the Add “Syst:Err?” Query option. If you make syntax errors, the errors will display in the Response field. The example in Figure 2-30 shows 0, “No error”. Different digits and error description will be the response if you make errors.

The Response field can also be used to query instrument settings. Try the following queries and check the response:

output?	will query output state
frequency?	will query frequency setting
volt?	will query amplitude setting

The commands as you type them will be saved as an history list. You can use any command from this list by unfolding the list and selecting a command, then clicking on Send to program this command.



*IDN?

☑ Add "Syst:Err? " query

□ Clear history list

Response

LeCroy,PXA125,0000201344,1.5;0,"No error"

Figure 2-30. Communicating with your Instrument

Chapter 3

ArbConnection©

What's In This Chapter

This Chapter contains information how to install, invoke and use ArbConnection. Introduction to ArbConnection and examples how to program instrument controls and parameters and how to generate waveforms and download them to the PXA125 are also given in the following sections.

Introduction To ArbConnection

In general, ArbConnection is a utility program that serves as an aid for programming the Model PXA125. ArbConnection has many functions and features of which all of them share a common purpose – controlling PXA125 functions from remote. As minimum, to use ArbConnection, you'll need the following tools:

1. Computer, Pentium II or better
2. Windows 95, or higher
3. High resolution screen, at least, 1024 x 768 pixels
4. Pointing device, mouse or ball
5. PXI chassis and installed PXA125
6. Last, but not least, some basic knowledge on how to operate computers and Windows-based programs.

ArbConnection operation is divided into three main functions: 1) Front panel control, 2) Waveform generation and editing and 3) FM waveform generation and editing. These operating options are described in this chapter. But, you must install ArbConnection before you can use it, so the next paragraphs describe installation and first steps before going into in-depth operation.

Installing ArbConnection

ArbConnection installs as part of the general installation process of the instrument drivers and programs. You can also install ArbConnection alone, in case you are not interested in installing PXI drivers but only want to test or explore its operation. The complete installation process is outlined in Chapter 2. To install ArbConnection without other parts of the installation package, proceed with Custom installation and check the ArbConnection option only, as shown in Figures 2-20 and 2-22

Quitting ArbConnection

Before you start roaming through menus and editing commands, we strongly recommend that you make yourself familiar with ArbConnection basics and concept. For now quit the program and spend some more time with this section of the manual. Point the mouse cursor to the File menu and press the left mouse button. Move the mouse cursor to the Exit command and press the left mouse button.

For the New and Advanced Users

For the New User

Learning to use ArbConnection is easy, intuitive and quick, even if you have never used such programs before. After you have installed ArbConnection on your computer read the following paragraphs to learn how to find your way around ArbConnection's menus.

Once you are familiar with the basics, you'll continue to learn about features, programming, and editing commands. If you can't find the answer to a question in this guide, call your distributor or the LeCroy customer support service near you and we'll gladly assist you with your problems.

For the Advanced User

If you are already familiar with computer conventions and have basic knowledge of Windows programming, you may want to skip some of the following paragraphs.

Conventions Used in This Manual

This manual uses certain typographical conventions to make it easier for you to follow instructions. These conventions are described in the following:

[Enter, or ↵] Press the Enter or Return key.

[Esc] Press the Escape key.

[Alt-F] Press the Alt key and the key that follows, simultaneously. In this example the key that follows is F.

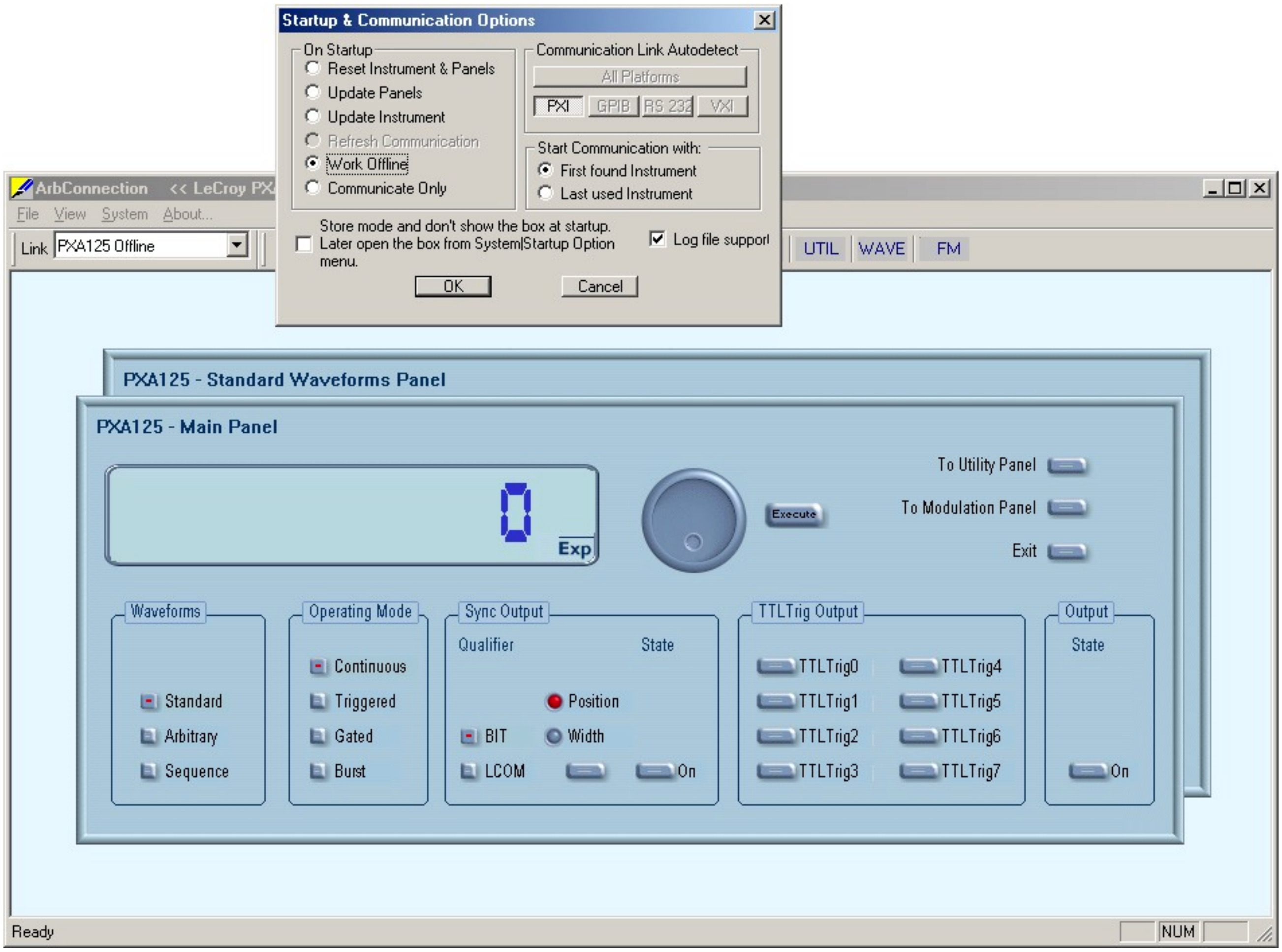
[Ctrl-S] Press the Control key and the letter that follows, simultaneously. In this example, the letter is S. The control key also appears in the menus as a target sign.

[↑] [↓] [→] [←] Press the Arrow key with the symbol pointing in the direction specified (i.e., up, down, left, or right).

<+> Press the key for the character or word enclosed in angle brackets. In this case, the Plus sign key.

The Opening Screen

Invoke ArbConnection by double clicking on the icon. If you cannot find the icon on your desktop, click on Start, Programs and ArbConnection. The opening screen will show. If you installed the program correctly, your screen should look as shown in Figure 3-1.



Startup & Communication Options

On Startup
○ Reset Instrument & Panels
○ Update Panels
○ Update Instrument
○ Refresh Communication
● Work Offline
○ Communicate Only

Communication Link Autodetect
All Platforms
PXI GPIB RS 232 VXI

Start Communication with:
● First found Instrument
○ Last used Instrument

□ Store mode and don't show the box at startup. Later open the box from System|Startup Option menu.

☑ Log file support

OK Cancel

Figure 3-31. The opening Screen

The Startup & Communication Options dialog box is displayed at the top. You can check the “Store and don’t show…” so next time you invoke ArbConnection, this dialog box will not be displayed. The purpose of this dialog box is to update the program in the way you intend to use it. Since ArbConnection is used for other platforms, it is extremely important to make sure that the selected communication link for the PXA125 is PXI so the first thing to do is click on the PXI button to activate this link. Other options are GPIB, RS232 and VXI but they should be disabled for now. If you press the All Platforms button, then the program will attempt to connect every possible link, which usually takes much longer.

The On Instrument Activation group lets you select the opening scenario for the instrument. The various options are described below:

Reset Instrument and Panels – will reset the generator and the control panels to factory default values. Parameter defaults are listed in Chapter 4 of this manual.

Update Panels – can be used in case you had to leave the program but left the generator still operating. The update Panels option will query instrument setting and will update the control panels accordingly. Note that waveforms cannot be read from the instrument so only control panels will be affected from this operation. The Update Panels option is useless if you powered down the host computer.

Update Instrument – can be used every time you invoke ArbConnection. Every time you exit ArbConnection, an image of the last control panel setting is automatically stored on your hard drive so the next time you invoke the program the control panels are updated from this image. When you use the Update Instrument option, the instrument will update its operational settings and parameter values only but will not download waveforms.

Refresh Communications – will activate the Autodetect engine and will attempt to reconnect all devices that respond to ArbConnection calls. The use of this option is normally not required except in isolated incidents that you suspect that ArbConnection has not detected all instruments that are present in the chassis.

Work Offline – will let you use ArbConnection without actually controlling or having an active instrument plugged in. This option also lets you prepare and save control panel settings for later use.

Communicate Only – will make the connection between ArbConnection and the instrument but will not synchronize between the two. You can use this option if you already modified instrument parameters but had to leave the program for a different application, otherwise, this option It is not recommended for normal use.

There are two more options to select from this dialog box. The Start Communications with:

First Found instrument – will start ArbConnection communication with the first instrument that was detected. If there are more than one instrument plugged into the chassis, you may want to select the next option.

Last Used Instrument – will direct ArbConnection communications to the instrument you used the last time before you had to exit the program.

If you chose to hide this dialog box, you can still access and change the options from the System command, at the top of the screen.

Make your selection and click OK. The Startup & Communication Updater dialog box will be removed from the screen. And two panels: the Main and the Standard Waveform panels will now be accessible. But before we go into panel operation, lets look at the toolbars at the left top of the screen as shown in Figure 3-2.

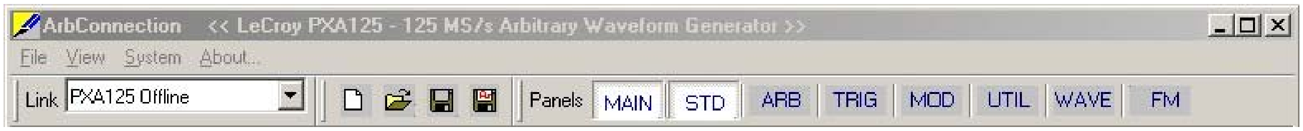

ArbConnection << LeCroy PXA125 - 125 MS/s Arbitrary Waveform Generator >>
File
View
System
About...
Link
PXA125 Offline
Panels
MAIN
STD
ARB
TRIG
MOD
UTIL
WAVE
FM

Figure 3-32. ArbConnection's Toolbars

The standard Windows **Menu Bar** is the top bar. It provides access to main system controls like saving files, and viewing or removal of screen images. Some of the File commands such as New, Open, Save and Save As… are also available as icons.

The second bar is called **Active Instrument** bar. It provides direct access to different instruments that are plugged into the PXI chassis. ArbConnection can operate many PXA125's units simultaneously. If an instrument was connected to the PXI bus while invoking ArbConnection, it will automatically be detected by the program and its link location and identification displayed in the Link field.

HINTS

1. **Active instrument means that changes you make on panels and waveform composers will affect this instrument only.**
2. **ArbConnection can address any instrument, which is plugged into your PXI system.**
3. **ArbConnection can address any combination**

of instruments as long as they are detected and displayed in the Active Instrument field.

The **Panels** bar provides direct access to instrument control panels. The individual control panels are explained later in this chapter. The MAIN, STD, ARB, TRIG, MOD and UTIL buttons will bring up to the screen panels that are associated with these names. The WAVE and FM buttons will open the waveform and FM waveform composers. The opening screen has two panels – MAIN and STD so the two buttons on the Panels bar are depressed. Click on other buttons and interactively get the feel how ArbConnection opens and cascades panels.

ArbConnection Features

In general, ArbConnection has one basic purpose – controlling PXA125 functions and parameters. On the other hand, the PXA125 has three main features: 1) Generating standard waveforms, 2) Generating arbitrary waveforms and 3) Generating modulated waveforms. So, ArbConnection had to be constructed in such a way to provide complete control capability from the computer screen.

ArbConnection has three main screens: 1) Control panels, 2) Waveform composer and 3) FM waveform composer. The various screens along with instructions how to access and use them are described below in detail.

The Control Panels

The control panels look and feel just as if you would operate an instrument from its front panel. They even look like a front panel of an instrument, so operating function and changing parameters are very intuitive. Lets look at the first panel that shows at the opening screen. This panel, as shown in Figure 3-3, is called the Main Panel.

To begin with, lets explore the panel controls to see how they feel, react and what they do. All other panels share almost the same feel, so the description of how to operate the Main Panel can serve as general guide for controlling the rest of the panels.

Looking at the panel you can identify the following controls: Push buttons, LED's, radio buttons, Dial and Digital display. The function of each control is described below.

Push Buttons – These are used for two functions: a) For selecting an active parameter and b) for toggling a function on and off. For example, the button in the Parameters group will select between the Amplitude and Offset LED's. The selection buttons have momentary push action, so pressing (clicking a mouse on) this button will change the reading on the display to correspond with the selected parameter.

The State button in the Output group has toggles the output on and off. The first mouse click will push the button inwards and will turn on a red bar at the center of the button, indicating that the function is on. The second mouse click will turn the function off.

Radio Buttons – Are used for changing operating modes, or selecting between mode options. One of the radio buttons is always on with a red dot in its center, indicating its state condition. Pressing some radio buttons will pop up a new panel. For example, if you press the Triggered button in the operating Mode group, triggered operation will be enabled and the Trigger Panel will automatically pop up and cascade in front of the Main Panel.

LED's – The LED's indicate which of the parameters are displayed on the Digital Display. Red LED indicates that the parameter name next to this LED is selected. Only one LED can be ON at a time.

HINT

LED's may be turned on by pressing the button below. You can also select an LED by clicking on the LED itself!

Dial – Use the dial to modify displayed reading. To use the dial, press and hold the mouse cursor on the dial and move the mouse in a clockwise circle to increase the number, or counterclockwise circle to decrease the displayed number. The dial modifies digits at the cursor position and will allow modification within the legal range of the displayed parameter. If you reach the end of the range, the dial will have no further effect on the display. If you do not want to use the dial, you can still change the display reading by using the [↑], or [↓] keys, or simply type the required number using the your standard keyboard features.

NOTE

After you change the displayed readout, the PXA125 will be updated with the new parameter only after you press the Execute button.

Digital Display – The display is used for displaying and reading various PXA125 parameters, just as you would use it on your instrument.

The Main Panel

The Main Panel, as shown in Figure 3-3, is the first panel you see after invoking ArbConnection. If you press the Exit button on the main panel, you'll be exiting the program completely. Notice how buttons and LED's are grouped; this is done specifically so that common parameters are placed in functional groups. The Main Panel groups allow (from left to right) adjustment of amplitude and offset, selecting the waveform type, selecting the operating mode, controlling the SYNC output, controlling the backplane trigger line outputs and turning ON, or OFF the main output.

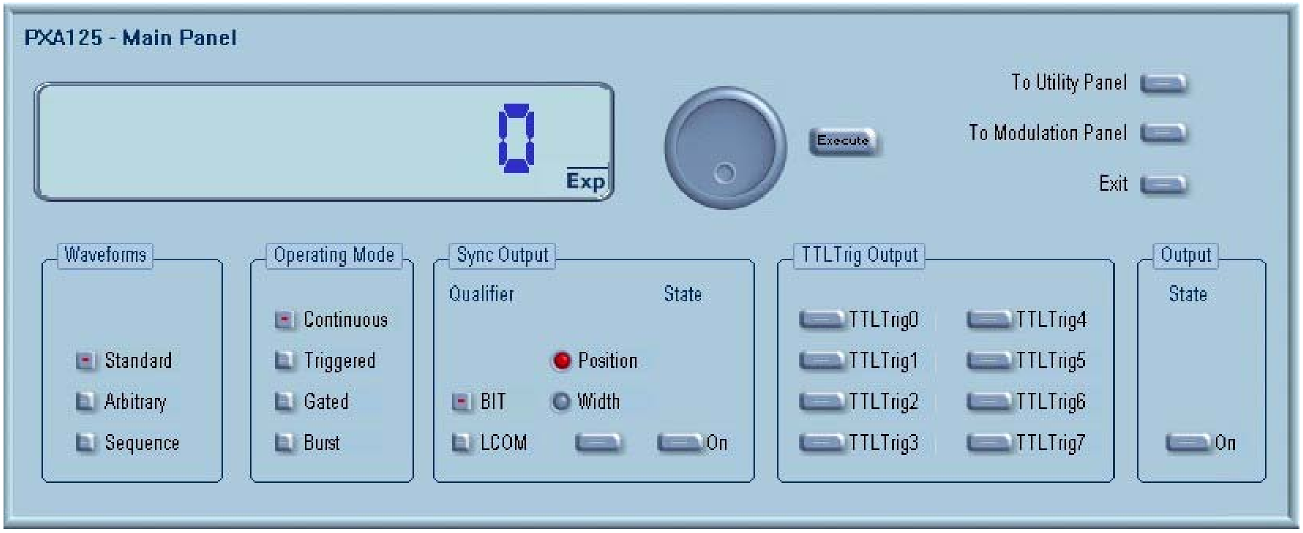


Figure 3-33. The Main Panel

If you are connected properly to the PCI bus and ArbConnection has detected this link and placed the address in its Link field, then every time you press a button, you are getting an immediate action on the PXA125. It is different if you are changing parameters on the display; Doing this, you'll have to press the Execute button for the command to update the instrument.

HINT

If your instrument has not been detected automatically, or if you connected the PXA125 only after you invoked the program, you can still detect your instrument by selecting the Refresh Communication from the System command. You can always work offline if you just want to explore

the program.

The functional groups in the Main Panel are explained below.

Waveforms The Waveform group is used for selecting between waveform types. The PXA125 provides three types of waveforms: Standard, Arbitrary and Sequenced. Pressing one of these buttons will cause two reactions: 1) The PXA125 will change the output waveform to the selected option and 2) Another panel, associated with the selected waveform type will be invoked and brought to the front. The default waveform type is Standard. Therefore, the opening screen has this panel behind the Main Panel. If you want to change standard waveform parameters, you can either click on the standard waveform panel, or select this panel from the Panels bar.

Operating Modes The Operating Modes group is used for selecting the active trigger mode for the instrument. You can select between continuous, triggered, gated and burst modes. There is no additional panel associated with the continuous mode, but if you press one of the other operating mode options, the PXA125 will be updated with the new trigger option and the Trigger Panel will pop up, allowing adjustments of trigger parameters.

SYNC Output The SYNC Output group controls SYNC parameters and enables or disables the SYNC output. The SYNC output is enabled when the State button is ON. The SYNC Width and Position parameters will affect the PXA125 when the BIT validator is selected. In LCOM mode, the width and the position will have no affect on the SYNC output.

TTLTrig Output The TTLTrig lines are available on a backplane (P2) connector, The function of the TTLTrig lines is very similar to the SYNC output. Every time the generator outputs a cycle, it generates a pulse from its front panel SYNC output connector. The same SYNC signal can be placed on one of the TTLTrig lines and can be used for triggering other instruments that have TTLTrig line capabilities. Enable one or more of these lines by pressing the appropriate buttons

Note

The SYNC parameters have direct effect on the TTLTrig line thus, by changing SYNC position, or width you are automatically assigning the same parameters to the backplane trigger lines.

Output The Output group is used for controlling the output state of the PXA125. The button toggles the output ON and OFF.

Besides the grouped buttons and parameters, there are three other buttons on the right top corner of the panel. These buttons will provide access to the (from top) Utility and Modulation Panels. Also, if you press the Exit button, the program will be terminated.

The Standard Waveforms Panel

The Standard Waveforms panel, as shown in Figure 3-4, is always invoked at the back of the Main Panel since the default output shape of the PXA125 is standard sine waveform. There are a number of ways to access this panel: You may click on it to bring it in front of the main Panel, you can click on the STD. button on the Instruments bar, or you can click on the Standard button within the Waveforms group in the Main Panel. The functional groups in the Standard Waveforms Panel are described below.

Parameters

The Parameters group contains two parameters: Amplitude and Offset. To access the required parameter, click on the button until the LED next to the required parameter turns on. The value that is associated with the lit LED is displayed on the digital display. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading.

Note

Normal color of the digital reading is dark blue. If you modify the reading, the color changes to a lighter shade of blue, indicating that the PXA125 has not been updated yet with the new parameter. Pressing Execute will update the instrument and will restore the color of the digital readout to dark blue, indicating that the displayed value is the same as the generator setting.

Also note that the digital readout has an autodetect mechanism for the high and low limits. You cannot exceed the limits if you are using the dial but only if you use the keypad. In case you do, the program will not let you download an illegal parameter and you'll be requested to correct your setting.

Frequency

The button in the Frequency group lets you program the output frequency for the selected waveform shape. The frequency parameter may be modified when the LED illuminates. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading.

Note

Normal color of the digital reading is dark blue. If you modify the reading, the color changes to a lighter shade of blue, indicating that the PXA125 has not been updated yet with the new parameter. Pressing Execute will update the instrument and will restore the color of the digital readout to dark blue, indicating that the displayed value is the same as the generator setting.

Also note that the digital readout has an autodetect mechanism for the high and low limits. You cannot exceed the limits if you are using the dial but only if you use the keypad. In case you do, the program will not let you download an illegal parameter and you'll be requested to correct your setting.

Besides the grouped buttons and parameters, there are two other buttons on the right top corner of this panel. These buttons will provide back access to the Main Panel and will also allow removal of the Waveforms Panel from the screen.

10 MHz Ref The 10 MHz Ref group provides selection between internal or external references. The PXA125 has three reference options:

Internal – connects the CLK10 reference from P2 to the sample clock generator. The accuracy and stability of the CLK10 is around 100ppm.

Note

Compact PCI chassis do not have P2 connectors on their backplanes. If you have a CPCI chassis and intend to use the PXA125, you have two options: 1) Apply a 10 MHz reference to the 10 MHz Ref input, or 2) Use the TCXO setting with your card. If you fail to provide an appropriate reference signal, the instrument will not function properly.

External – connects the front panel 10MHz Reference Input to the sample clock generator. The accuracy and stability of this input is determined by the source, which you apply to this input.

TCXO – connects a 1ppm reference to the sample clock input. Click on the Internal button to select the Internal CLK10 reference, the External button to activate the external reference input, or the TCXO to select the 1 ppm reference.

⚠ ***CAUTION***

By selecting an external reference you are disabling the internal reference circuit. If you do not have a 10 MHz reference connected to the instrument, the output will generate erroneous frequencies.

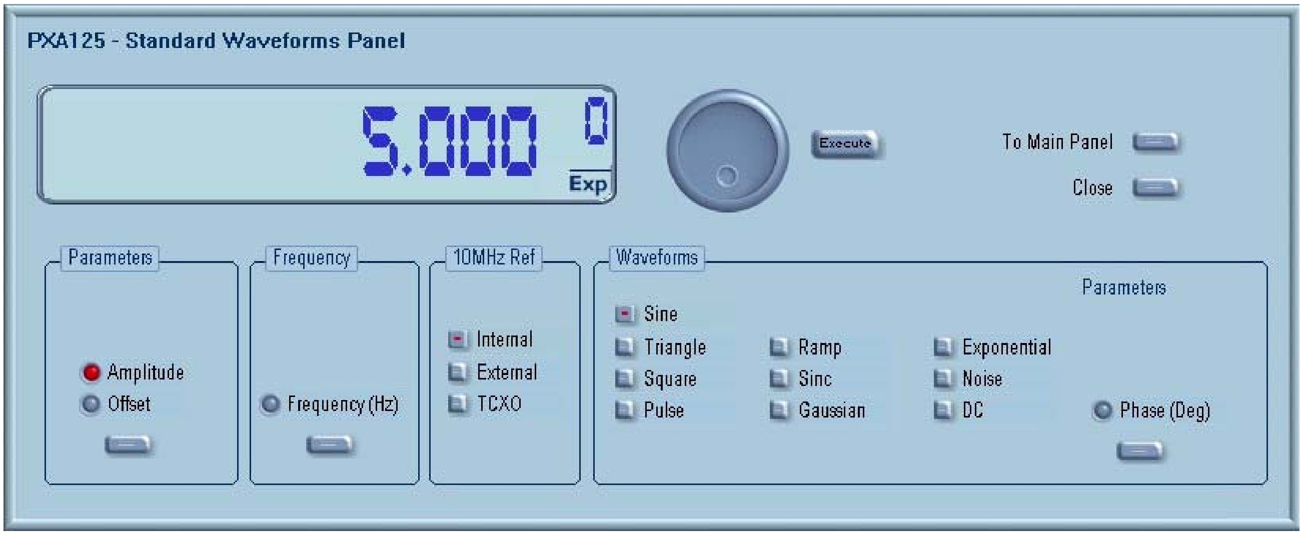


Figure 3-34. The Standard Waveforms Panel

Waveforms The Waveforms group lets you select standard functions and modify their associated parameters. You can select any of nine standard wave shapes and the PXA125 will generate them for you. Select your standard output waveform by pressing one of the buttons. After you have made this selection, you'll be granted access to parameters that are unique for this function. For example, if you click on the Square button, the Duty Cycle parameter will be available for modification. Likewise, if you select the pulse shape, the fall, high, rise and delay parameters will be available for modification. Note that by clicking a button in this group, you are immediately updating the PXA125 output with this waveform shape.

The Arbitrary & Sequence Panel

The Arbitrary & Sequence panel, as shown in Figure 3-5, is invoked either by pressing the ARB button on the Panels bar, or clicking on the Arbitrary or Sequenced Waveforms on the Main Panel. Note that if you invoke the Arbitrary & Sequence Panel from the Panels menu, the PXA125 will not change its output type. On the other hand, if you select the arbitrary, or the sequenced options from the Main Panel, besides invoking the Arbitrary & Sequenced Panel, the PXA125 will immediately change its output type to the selected waveform type. The functional groups in the Arbitrary Waveforms Panel are described below.

Parameters The Parameters group contains parameters for adjusting Amplitude and Offset. Actually, the values exhibited in this group are exactly the same as in the Main Panel, so every time you change amplitude and offset in the Parameters group, the other panels are updated automatically. You can also select an active segment from this group.

To access the required parameter, click on the button below until the LED next to the required parameter turns on. The value that is associated with the lit LED is displayed on the digital display. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading.

Note

Normal color of the digital reading is dark blue. If you modify the reading, the color changes to a lighter shade of blue, indicating that the PXA125 has not been updated yet with the new parameter. Pressing Execute will update the instrument and will restore the color of the digital readout to dark blue, indicating that the displayed value is the same as the generator setting.

Also note that the digital readout has an autodetect mechanism for the high and low limits. You cannot exceed the limits if you are using the dial but only if you use the keypad. In case you do, the program will not let you download an illegal parameter and you'll be requested to correct your setting.

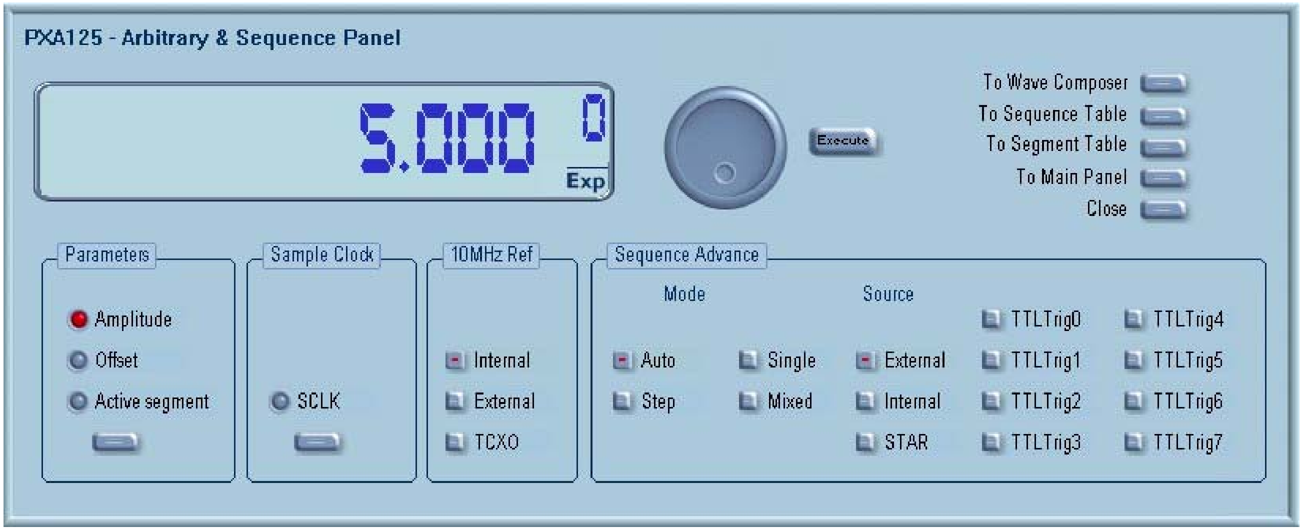


Figure 3-35. The Arbitrary & Sequence Panel

Sample Clock The Sample Clock group is comprised of one parameter that controls the sample clock frequency. Note that the sample clock rate is programmed in units of S/s (samples per second) and will affect the Model PXA125 only when it is programmed to output arbitrary or sequenced waveforms.

To access the required parameter, click on the button until the LED next to the required parameter turns on. The value that is associated with the lit LED is displayed on the digital display. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading.

10 MHz Ref The 10 MHz Ref group is a duplicate of the same group in the Main Panel. The 10 MHz Ref group provides selection between internal (CLK10), external, or TCXO references. Click on the Internal button to select the Internal CLK10 reference, the External button to activate the external reference input, or the TCXO button to select the better accuracy and stability option.

CAUTION

By selecting an external reference you are disabling the internal reference circuit. If you do not have a 10 MHz reference connected to the instrument, the output will generate erroneous frequencies.

Sequence Advance The Sequence Advance group provides control over advance modes for the sequence generator. Advance options are: Auto, Stepped, Single and Mixed. Refer to the PXA125 operation instructions to find out more when and how to use these advance modes. You should be careful while selecting modes because it is possible to cause settings conflict, for example, if you select the Single option and you forgot to change your trigger mode to Triggered.

The Sequence advance group lets you also select the source for advancing the sequence steps. Description of the various advance sources is given below.

External – selects the TRIG IN input as the advance source. Note that this option will affect the generator only when it is placed in operational mode that requires trigger stimuli.

Internal – disables the TRIG IN input and routes an internal trigger generator to the trigger advance circuit.

STAR – connects the trigger source to a backplane STAR line. This line serves as common trigger line to all modules.

TTLT0 to TTLT7 – connects the trigger source to one of eight backplane trigger lines.

Besides the grouped controls, there are five other buttons available on the Arbitrary & Sequence Panel. These buttons provide access to (from top to bottom) Wave Composer, Sequence Table, Segment Table, Main Panel. The last button removes the Arbitrary & Sequence Panel from the screen. The Wave Composer will be discussed separately in this chapter. The Segment and Sequence Tables are described below.

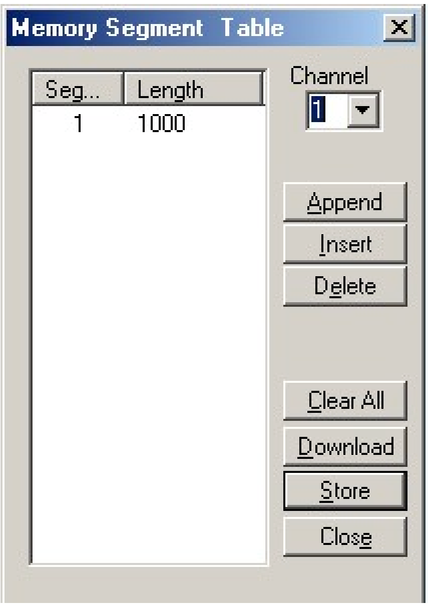
Using the Segment Table

If you want to learn more about segment control, you should refer to Chapter 4 of this manual. In general, the PXA125 can generate arbitrary waveforms but before it can generate anything, the waveforms have to be downloaded to the instrument from a host computer. The waveforms are downloaded as coordinates and are stored in the PXA125 is a place designated as “waveform memory”. The waveform memory has a finite size of 2Meg.

Having such a long memory to store waveforms does not necessarily mean that you have to use the entire memory every time you download a waveform. On the contrary, the PXA125 allows segmentation of the memory so that up to 4096 smaller waveforms could be stored in the waveform memory. There are two ways to divide the waveform memory to segments: Define a segment and load it with waveform data, define the next and load with data, then the third etc. The second alternative is to use what ArbConnection has to offer and that is to make up one long waveform that contains many smaller segments, download it to the instrument in one shot and then download a segment table that splits the entire waveform memory into the required segment sizes.

Want to use it? Here is how it is done.

Click with your mouse on the To Segment Table button. A dialog box as shown in Figure 3-6 will pop up.



Seg...	Length
1	1000

Channel: 1

Figure 3-36. The Segment Table

The **Channel** field lets you select the active channel. If "1" is displayed, then the table you'll be constructing will affect channel 1. Likewise, select "2" if you want to program the table for channel 2.

The two main fields in the segment table are Seg and Length. The **Seg** (segment) is an index field, which can have values only, from 1 to 4096. The **Length** is always associated with the segment number. You can program any segment length from 16 to 4 Meg provided that the sum of the length for all programmed segments does not exceed your purchased waveform memory size.

Use the **Append** key to add a segment at the end of the segment list. Use the **Insert** key to insert a segment at the cursor location. The **Delete** key is used for deleting a segment at the cursor position.

The **Clear All** key will remove all segments from the table and will let you start designing your segment table from fresh.

Click on the **Close** to discard of the contents of the dialog box without saving your last actions and to remove the Segment Table from the screen but click on the **Store** key if you want just to save your work before you close the dialog box.

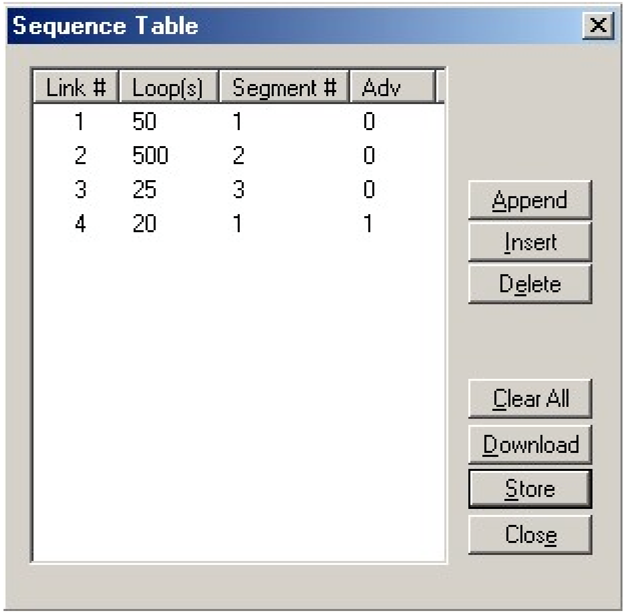
The **Download** key has double action, it will download the segment table to the instrument and will store the contents of your segment table.

Using the Sequence Table

If you want to learn more about sequences and sequence control, you should refer to Chapter 4 of this manual. In general, the PXA125 can generate sequenced waveforms but waveforms must be loaded to the instrument from a computer before it can generate sequences. The waveforms are downloaded as coordinates and are stored in the PXA125 is a place designated as “waveform memory”. The waveform memory has a finite size of 2Meg.

As was explained in Using the Segment Table above, the waveform memory can be divided into smaller segments and up to 4096 segments can be defined and used as individual arbitrary waveforms. Having a limited size of waveform memory can, for some applications, pose a limitation however, if sections of the waveform are repetitive, one may use the sequence generator to take these segments and replay them as part of the complete waveform without loosing valuable memory space and without scarifying waveform coherences, or integrity. The tool for using repetitive and multiple segments in one long waveform is called Sequence Generator. ArbConnection has a special dialog box where sequences are designed. This tool is called – Sequence Table.

The Sequence Table is demonstrated in Figure 3-7. To access this table click on the “To Sequence Table” button. Description of the various controls in this table is given below.



Sequence Table

Link #	Loop(s)	Segment #	Adv
1	50	1	0
2	500	2	0
3	25	3	0
4	20	1	1

Figure 3-37. The Sequence Table

There are four major elements that you should consider while programming a sequence table. They are: Link#, Loop(s), Segment# and Adv. These elements are explained below.

Link # – defines an index array for the sequence table. When generating sequences, the instrument steps though the link # in descending order therefore, make sure that you enter your waveform segments in exactly the order you would like them at the output.

Loop(s) – define how many times the segment will loop for the selected link. For example, if you program 2, the waveform will cycle twice through the same segment before transitioning to the next link.

Segment # – associates waveform segments with links. You can use different segments for different links or you can use the same segment for a number of links. There are no limitations how you associate links to segments, except you cannot program in the sequence table segments that were not defined earlier.

Adv – is a special code that is used in conjunction with the mixed advance mode. This bit flags the PXA125 if the selected link is continuous or stepped. Information on the Mixed sequence advance mode is given later. “0” flags continuous, “1” flags stepped.

Figure 3-7 shows an example of a 4-step sequence of which the first waveform is made of segment 1, which will loop 20 times; segment 2, looping 500 times; segment 3 looping 25 times and segment 4, which will loop 20 times. Also notice that mixed mode advance flag has been assigned to segment 4. This flag will cause the sequence to stop on segment 4 until the PXA125 will be triggered for the next sequence cycle.

There are control buttons on the right of the sequence table. Use the **Append** key to add a step at the end of the sequence list. Use the **Insert** key to insert a step at the cursor location. The **Delete** key is used for deleting a step at the cursor position.

The **Clear All** key will remove all steps from the table and will let you start designing your sequence table from scratch.

Click on the **Close** to discard of the contents of the dialog box without saving your last actions and to remove the sequence Table from the screen but click on the **Store** key if you want just to save your work before you close the dialog box.

The **Download** key has double action, it will download the sequence table to the instrument and will save the contents of your table so the next time you open this table, it will have the same contents as you saved in your previous session.

The Trigger Panel

The Trigger panel, as shown in Figure 3-8, is invoked either by pressing the TRIG button on the Panels bar, or clicking on one of the Triggered, Gated, or Burst buttons on the Main Panel. Note that if you invoke the Trigger Panel from the Panels menu, the PXA125 will not change its operating mode. On the other hand, if you select the Trigger option from the Main Panel, besides invoking the Trigger Panel, the PXA125 will immediately change its operating mode to triggered. The functional groups in the Trigger Panel are described below.

Trigger Parameters

The Trigger Parameters group contains two parameters for each channel: Phase and Count. To access the required parameter, click on the button below the Channel 1 or Channel 2 group until the LED next to the required parameter turns on. The value that is associated with the lit LED is displayed on the digital display. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading.

Note

Normal color of the digital reading is dark blue. If you modify the reading, the color changes to a lighter shade of blue, indicating that the PXA125

has not been updated yet with the new parameter. Pressing Execute will update the instrument and will restore the color of the digital readout to dark blue, indicating that the displayed value is the same as the generator setting.

Also note that the digital readout has an autodetect mechanism for the high and low limits. You cannot exceed the limits if you are using the dial but only if you use the keypad. In case you do, the program will not let you download an illegal parameter and you'll be requested to correct your setting.

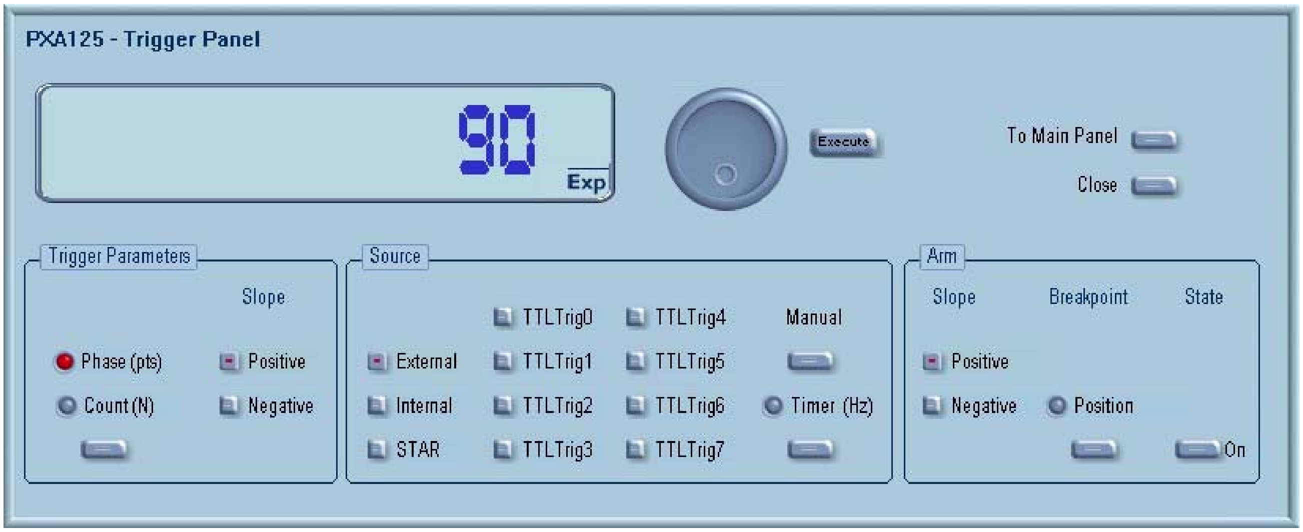


Figure 3-38. The Trigger Panel

Slope The Slope group lets you select edge sensitivity for the trigger input of the PXA125. If you click on Positive, the instrument will trigger on the rising edge of the trigger signal. Likewise, if you click on Negative, the instrument will trigger on the falling edge of the trigger signal.

Source The Source group provides access to trigger source parameters. Click on External to enable the rear-panel TRIG IN connector. If you select the external option, you must connect your stimulating signal to the trigger input. You can also use the Manual button to trigger the instrument. If you select the Internal option, you will be replacing the external signal with an internal, free-running, synthesizer that is connected internally to the trigger input. The period of the internal synthesizer is programmable using the Timer button in this group.

Arm

The Arm group is really a semi-orphan function on the Trigger Panel. This is the only function that actually will work in continuous mode however, since the arm commands cause interruption of the PXA125 output and arm control is done with the trigger input, it was placed in the trigger section of ArbConnection. While modifying Arm commands, bear in mind that the instrument must be in continuous mode only.

The arm function requires definition of breakpoints. To turn ARM on and off, click on the **State** button. Select the Position LED to program **breakpoint** position. The **Positive** on **Negative** slope options will determine edge sensitivity for the trigger input.

There are two additional buttons at the right top corner of the Trigger Panel. You can close the Trigger Panel by clicking on the Close key, or return to the main panel and leave the Trigger Panel in the background if you click on the “To Main Panel” key.

The Modulation Panel

The Modulation panel, as shown in Figure 3-9, is invoked either by pressing the MOD button or “To Modulation Panel” button on the Main Panel. The modulation panel provides access and control for a number of functions: FM (frequency modulation), FSK (frequency shift keying) and Sweep. The PXA125 can perform only one of these functions at a time and ArbConnection will generate an error if you try to change the state to on for more than one modulation option.

The functional groups in the Modulation Panel are divided to three parts: Frequency Modulation, FSK and Sweep. These groups are described below.

Frequency Modulation

The Frequency Modulation group contains parameters for controlling the sample clock. To turn the FM function on and off, click on the State button in this group. There are sub-groups in the FM block: Parameters, Shape, Mode and Slope.

Shape – Determines how the frequency will change. There are four standard patterns that are built into the instrument: sine, triangle, square and ramp, which could be selected from a built-in lookup table. The arbitrary waveform shape must be loaded to the instrument from an external application, or from the FM Composer. Click on the button next to the required modulating waveform shape to select it.

Parameters – Allow adjustment of waveform parameters that are selected from the Shape sub-group. Notice that the parameters are changing as you select different shapes. To access the required parameter, click on the button below parameters sub-group until the

LED next to the required parameter turns on. The value that is associated with the lit LED is displayed on the digital display. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading.

Note

Normal color of the digital reading is dark blue. If you modify the reading, the color changes to a lighter shade of blue, indicating that the PXA125 has not been updated yet with the new parameter. Pressing Execute will update the instrument and will restore the color of the digital readout to dark blue, indicating that the displayed value is the same as the generator setting.

Also note that the digital readout has an autodetect mechanism for the high and low limits. You cannot exceed the limits if you are using the dial but only if you use the keypad. In case you do, the program will not let you download an illegal parameter and you'll be requested to correct your setting.

Mode- The Mode sub-group let you select one of the FM operating modes. You have a choice of continuous, triggered and gated FM. Unlike the main trigger function, the trigger modes for the FM function operate in continuous mode. For example, if you place the instrument in triggered FM, the instrument will idle at the carrier frequency and execute one FM cycle when triggered. Refer to the appropriate section in this manual to learn more about triggered FM.

Slope – The Slope sub-group let you select edge sensitivity for the trigger input of the PXA125. If you click on Positive, the instrument will trigger on the rising edge of the trigger signal. Likewise, if you click on Negative, the instrument will trigger on the falling edge of the trigger signal.

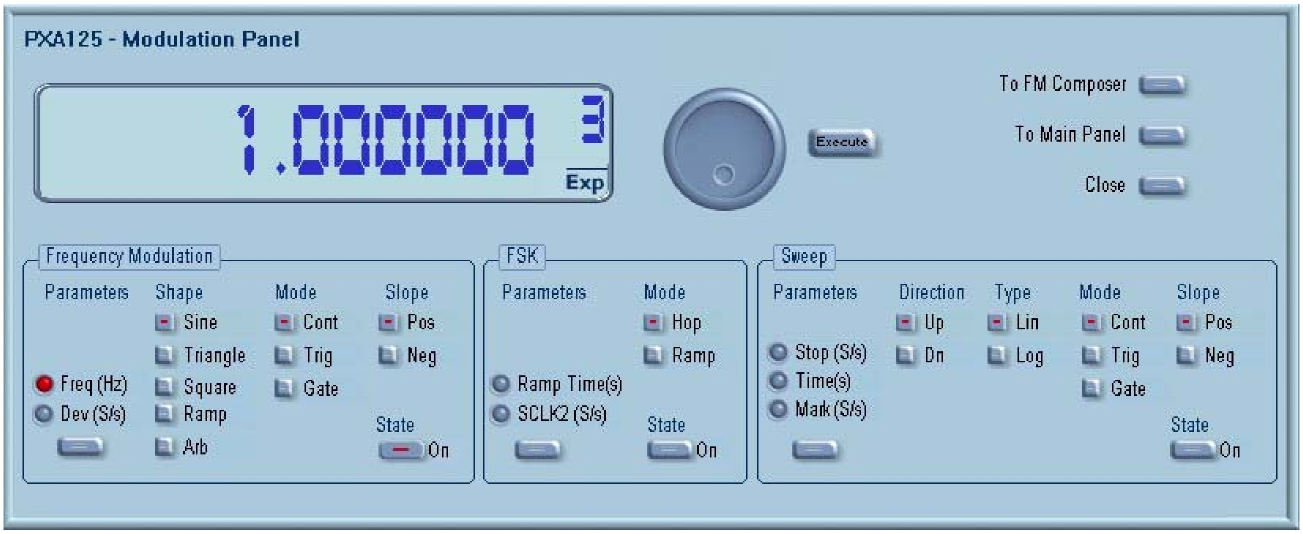


Figure 3-39. The Modulation Panel

FSK The FSK group contains parameters for controlling the FSK function. To turn the FSK function on and off, click on the State button in this group. There are two other sub-groups in the FSK block: Parameters, and Mode.

Mode – The Mode group lets you select between two shift options: hop and ramped. In hop mode, the sample clock hops to the shift frequency without delays. In ramped mode, the sample clock ramps to the shift frequency at a rate determined by the Ramp Time parameter. Click on the option you prefer to select between hop and ramped modes.

Parameters – Allow adjustment of Ramp Time and Shifted Sample Clock frequency parameters. To access the required parameter, click on the button below parameters sub-group until the LED next to the required parameter turns on. The value that is associated with the lit LED is displayed on the digital display. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading.

Sweep The Sweep group contains parameters for controlling sweep options. To turn the Sweep function on and off, click on the State button in this group. There are sub-groups in the Sweep block: Parameters, Direction, Type, Mode and Slope.

Parameters – Allow adjustment of Sweep Stop and Sweep Time. You can also place a marker at a position programmed by the Mark parameter. To access the required parameter, click on the button below parameters sub-group until the LED next to the required parameter turns on. The value that is associated with the lit LED is displayed on the digital display. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading.

Note

Normal color of the digital reading is dark blue. If you modify the reading, the color changes to a lighter shade of blue, indicating that the PXA125 has not been updated yet with the new parameter. Pressing Execute will update the instrument and will restore the color of the digital readout to dark blue, indicating that the displayed value is the same as the generator setting.

Also note that the digital readout has an autodetect mechanism for the high and low limits. You cannot exceed the limits if you are using the dial but only if you use the keypad. In case you do, the program will not let you download an illegal parameter and you'll be requested to correct your setting.

Direction – Use the keys in this sub-group to program sweep direction. Up select sweep from start to stop sample clock setting and Down selects sweep from the stop to start sample clock setting. Refer to Chapter 3 of this manual to learn more about sweep operation.

Type – Use these keys to select sweep step from two options: linear and logarithmic.

Mode- The Mode sub-group let you select one of the Sweep operating modes. You have a choice of continuous, triggered and gated sweep. Unlike the main trigger function, the trigger modes for the sweep function operate in continuous mode. For example, if you place the instrument in triggered sweep, the instrument will idle at the carrier frequency and execute one sweep cycle when triggered. Refer to the appropriate section in this manual to learn more about triggered sweep.

Slope – The Slope sub-group let you select edge sensitivity for the trigger input of the PXA125. If you click on Positive, the instrument will trigger on the rising edge of the trigger signal. Likewise, if you click on Negative, the instrument will trigger on the falling edge of the trigger signal.

There are three additional buttons at the right top corner of the Modulation Panel. You can close the Modulation Panel by clicking on the Close key, or return to the main panel and leave the Modulation Panel in the background if you click on the “To Main Panel” key. The “To FM Composer” button will invoke the FM composer program that will enable you to create and edit user-defined waveforms that will be used as modulating waveforms. The FM Composer program is described in greater details later in this chapter.

The Utility Panel

The Utility panel, as shown in Figure 3-10, is invoked either by pressing the UTIL button or “To Utility Panel” button on the Main Panel. The Utility panel provides access to general instrument controls that are not directly related to waveform generation. Through this panel you can control multiple instruments and/or multiple channels. If you made programming errors, you can use this panel to clear the error queue, or reset the PXA125 all together. The functional groups in the Utility are described below.

Multi-Instruments Synchronization Control

The Multi-Instrument Sync Control group lets you synchronize between multiple instruments and program phase offsets between instruments. Phase offset is applied between the leading edges of the two instruments and is given in units of waveform points.

To adjust the **Ph. Offset** parameter, click on the button below parameter. The LED next to the Ph. Offset turns on. The offset value is displayed on the digital display. You can use the dial, keyboard, or the [↑] [↓] keys to adjust the readout to the required setting. After you modify the reading, press Execute to update the PXA125 with the new reading. Note that after changing phase between instruments, you must re-activate the synchronization between the instruments, as explained below.

Set Configuration – The Set Configuration button provides access to a configuration table as shown in Figure 3-11. Use this table to configure which of the instruments will be master and which will be slaved to this master. The example in Figure 3-11 shows 5 instruments in the chassis of which the instrument in slot 4 was selected as master and slots 5 and 6 were programmed as slaves. There are some basic rules when selecting master and slave units as summarized in the following:

1. You must have minimum of two instruments in your chassis to be able to open and access the Multi-Instrument Synchronization Panel.

2. Actual slot location is critical for synchronizing instruments. Therefore, before you start synchronizing instrument, use the

Arb Detector utility to assign chassis and slot location. Information how to access and use the Arb Detector program is given in Chapter 2 of this manual. If you attempt to configure the synchronization table before assigning actual slot location, you'll get an error message as shown in Figure 3-12, which will advise you on the procedure to correct the situation.

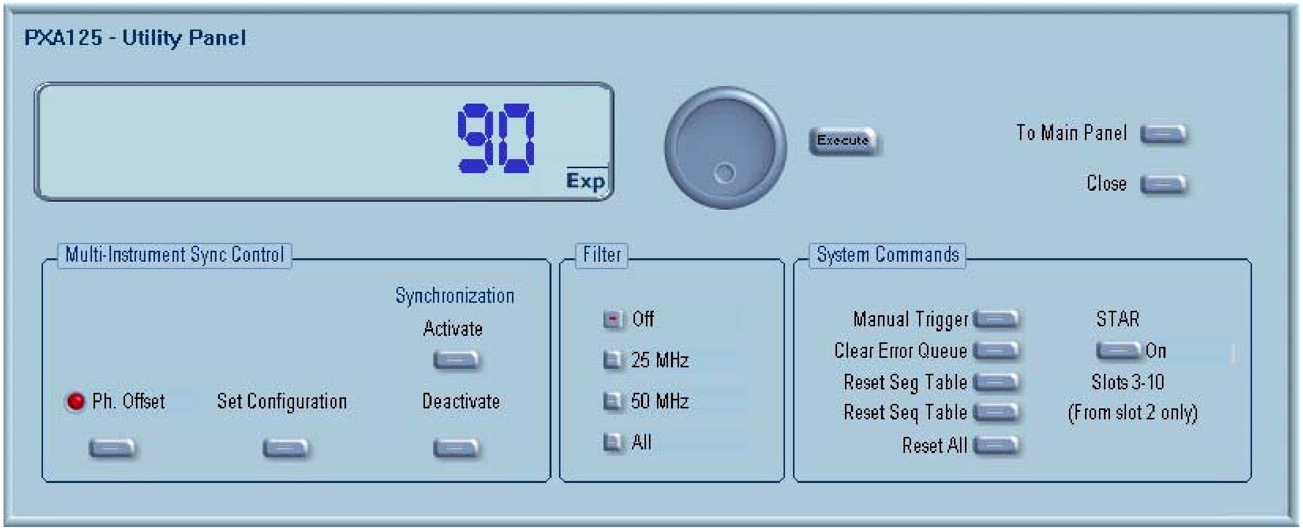


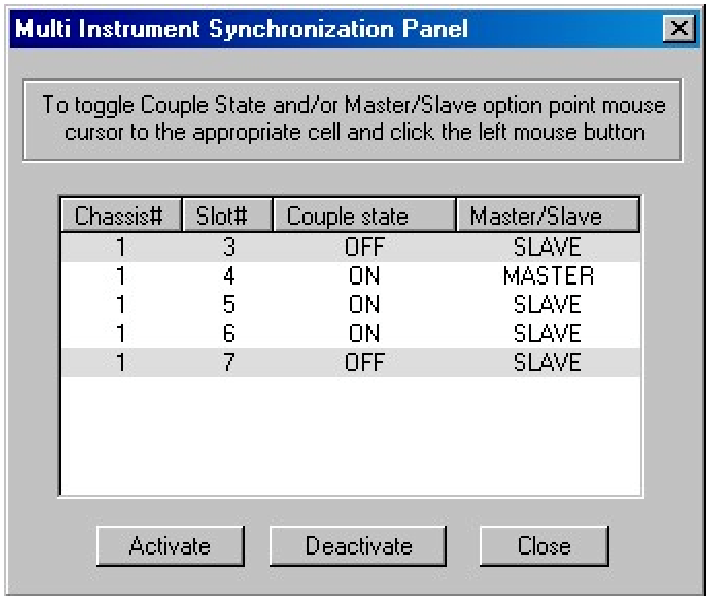
Figure 3-40. The Utility Panel

3. The master unit must always be at the left side of cluster of the synchronized instruments. Instruments plugged to the left of the master will automatically be excluded from synchronization and their Couple State set to OFF

4. One more or all instruments to the right of the master unit can be slaved. If you do not want to slave an instrument, you must configure its Couple State to OFF.

5. Master and Slave units must be adjacent. If you set the Couple State for one of the instruments OFF, then starting this slot, all instruments to the right will be removed from the cluster.

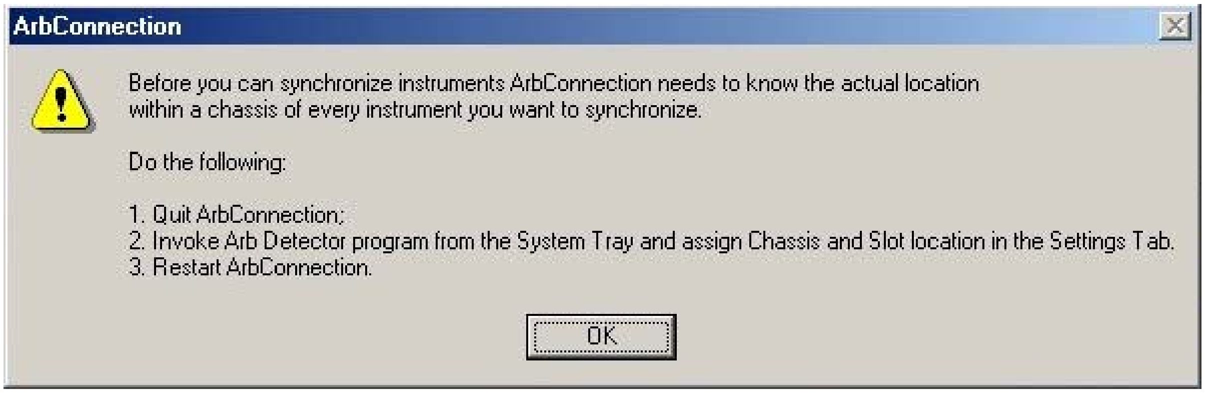
6. The Activate and Deactivate buttons are duplicated on the Utility Panel. After you configure the state of the instruments, you do not have to show this dialog box anymore but use the Utility Panel buttons.



To toggle Couple State and/or Master/Slave option point mouse cursor to the appropriate cell and click the left mouse button

Chassis#	Slot#	Couple state	Master/Slave
1	3	OFF	SLAVE
1	4	ON	MASTER
1	5	ON	SLAVE
1	6	ON	SLAVE
1	7	OFF	SLAVE

Figure 3-41. The Multi-Instrument Synchronization Panel



Before you can synchronize instruments ArbConnection needs to know the actual location within a chassis of every instrument you want to synchronize.

Do the following:

1. Quit ArbConnection;
2. Invoke Arb Detector program from the System Tray and assign Chassis and Slot location in the Settings Tab.
3. Restart ArbConnection.

Figure 3-42. Multi-Instrument Synchronization Error Message.

Activate – This button will activate synchronization between instruments as configured in the Multi-Instrument Synchronization Panel. If you pressed this button without first configuring master and slave units, the error message as shown in Figure 3-12 will be shown. The Activate button must be pressed after you change waveforms, phase offset, operation modes etc. You do not need to press this button if you change sample clock, amplitude and offset or other parameters that do not force waveform modification

Deactivate – This button will deactivate synchronization between instruments.

Filter The Filter group lets you place filters at the output path. There are two sets of filters – one for each channel. Click on 25 MHz, 50 MHz or All to select both. Click on Off to remove disable all filters.

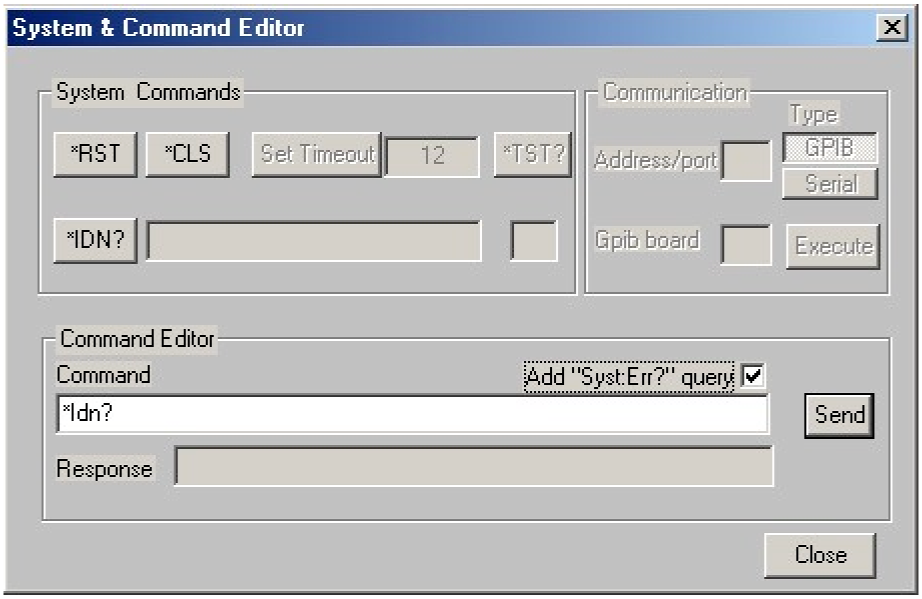
System Commands Commands in this sub-group will let you **Reset** the entire instrument to factory defaults, **Clear Error Queue**, if you made programming errors and want to remove the error list from the queue. You can also use the **Manual Trigger** key to trigger the PXA125 in case you do not have an external trigger stimuli connected to the TRIG IN connector. Other commands allow you to **Reset Segment Table** and **Reset Sequence Table**.

The System and Command Editor

ArbConnection requires certain conditions to work well with the PXA125. For example, The computer you are using must be connected to the PXI chassis through a known and well-defined communication link, such as, MXI3. Not only that the instrument must be connected but also, software and registers must be set properly for the generator to function properly. It is also possible that your computer is running smoothly but the PXA125 has problems communicating with your computer. The System and Command Editor dialog box has all the tools needed to make sure that the PXA125 is connected properly and that it responds correctly to machine commands.

Invoke the System and Command Editor from the System command at the top of the screen. Dialog box, as shown in Figure 3-13 will pop up. There are three main functions that you can execute using this dialog box: Set your communication link, reset and test your communication with the PXA125 and, last but no least, send SCPI commands to the instrument and see an immediate response. Description of the various elements of this dialog box is given below.

Communication The Communication group lets you select the proper communication link for connecting the PXA125 to the computer. In general, ArbConnection does that automatically for you when you invoke the program. This group is grayed out because it has no use for PXI



Set Timeout: 12

Add "Syst:Err?" query ☑

Command: *Idn?

Figure 3-43. System Control and the Command Editor

System Commands

The System Commands group lets you use common commands and test the PXA125 for proper operation. The Time Out field sets system timeout in units of seconds. The time out is required by the GPIB link in case your instrument does not communicate properly with on the bus. To avoid bus lockout, set the time out to small values. The default time out setting is 2. Note that PCI communication does not require time out setting therefore, it is blanked out for this instrument. The ***RST**, ***CLS** buttons are used for clearing error queue, or resetting the instrument to factory defaults.

The ***TST?** Invokes selftest routines. It is not relevant for the PXA125 and therefore blanked out in this dialog box. You should use this test when you suspect that something is wrong with your instrument. The PXA125 executes this test and generates a response that is placed in the field below the *TST? button. The response to the selftest command is numeral. Any response, besides “0”, means that your instrument failed the selftest. Additional information on the *TST? command is available in chapter 4 of this manual.

The *IDN? queries the generator’s identity. The response is placed in the field next to the *IDN? button. The response is organized into four fields, separated by commas, as shown below:

```
LeCroy,PXA125,0,1.0
```

Command Editor

Low-level SCPI commands and queries can be directly sent to the

PXA125 from the **Command** field and the PXA125 will response to queries in the **Response** field. The command editor is very useful while developing your own application. Build your confidence or test various commands using the command editor. This way you can assure that commands or syntax that you use in your application will respond exactly the same way as it responds to the editor commands. A complete list of SCPI commands is available in Chapter 4 of this manual.

The Waveform Composer

Being an arbitrary waveform generator, the PXA125 has to be loaded with waveform data before it can start generating waveforms. The waveform generation and editing utility is part of ArbConnection and is called – The Waveform Composer. This program gives you tools to create definitions for arbitrary waveforms. It can also convert coordinates from other products, such as, oscilloscopes and use them directly as waveform data. The program is loaded with many features and options so use the following paragraphs to learn how to create, edit and download waveforms to the PXA125 using the Waveform Composer.

You can invoke the Waveform Composer program from two places: On the Panels bar click on WAVE, or from the Arbitrary & Sequence Panel click on “To Wave Composer”. Figure 3-14 shows the waveform composer, immediately after you first invoke this utility.

The Wave Composer has three sections: Commands bar, Toolbar and Waveform screen. Refer to Figure 3-14 throughout the description of these parts.

The Commands bar

The commands bar provides access to standard Windows commands such as File and View. In addition, there are ArbConnection-specific commands such as Edit, Wave and System.

In general, clicking on one of the commands, opens a dialog box with an additional list of commands. Then, clicking on an additional command, may open a dialog box, or generate an immediate action. For example, Clicking on File and then Exit will cause an immediate termination of the Wave Composer. On the other hand, clicking on Wave and then on Sine, will open a Sine Wave dialog box that lets you program and edit sine wave parameters. The various commands in the Commands bar are listed and described below.

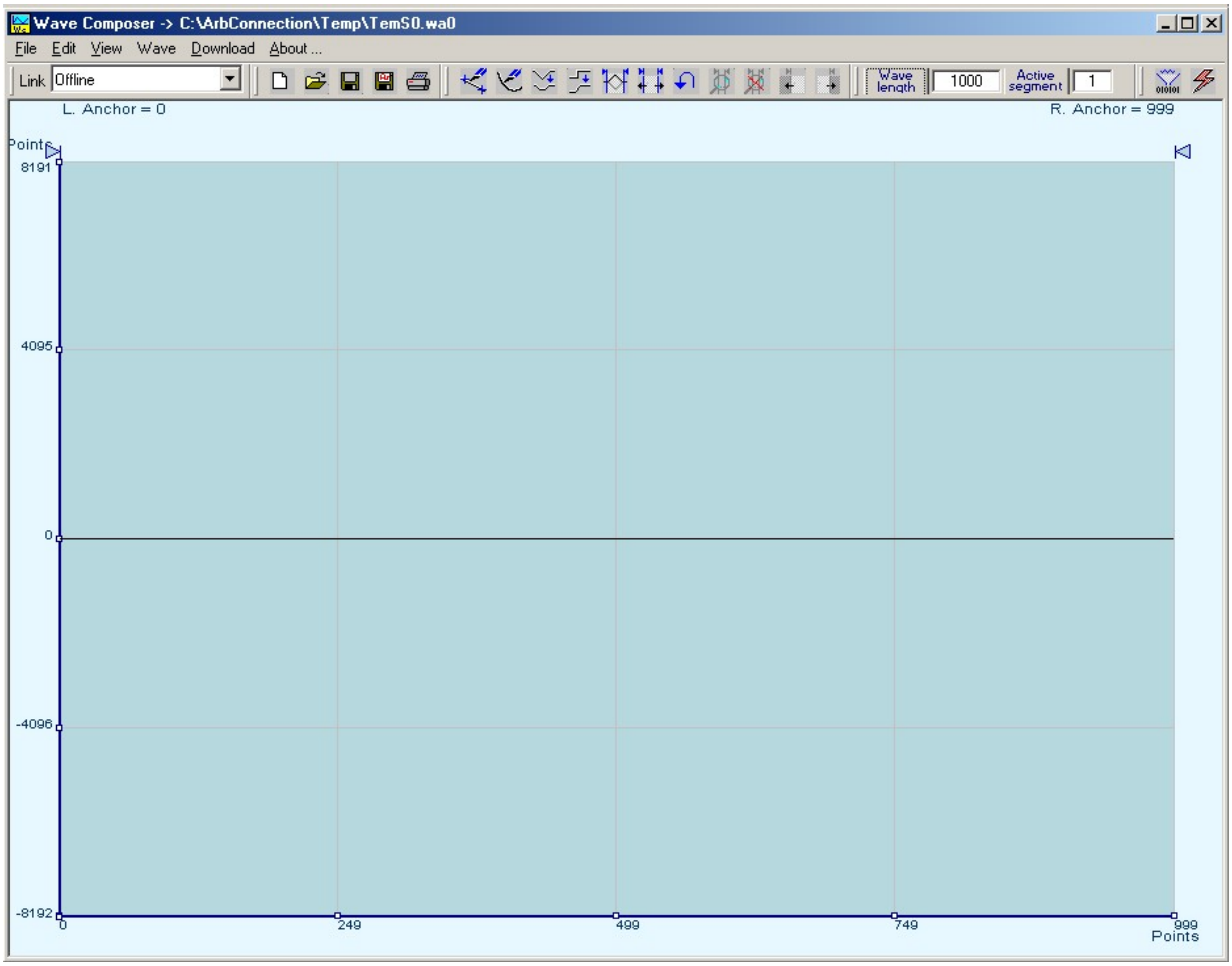


Figure 3-44. The Wave Composer Opening Screen

File Commands The File command has 4 commands that control waveform files. Also use this command to print the active waveform, or exit the wave composer program. Description of the various commands under File is given below.

New Waveform

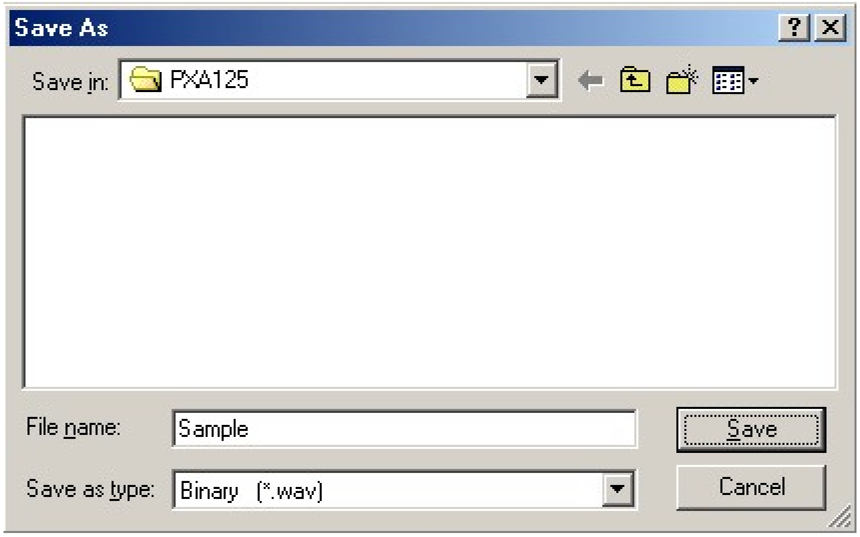
The New Waveform command clears current waveform from the screen. If you made changes to the waveform area and use the New Waveform command, you should save your work before clearing the screen. The New Waveform command is destructive to the displayed waveform.

Open Waveform…

The Open Waveform… command will let you browse your disk for previously saved waveform files and load these waveforms to the waveform area. This command is also very useful for converting waveform files to format that is acceptable by the Wave Composer. The Open Waveform command can convert ASCII. *CSV (comma delimited text), *PRN (space delimited text) and *.0* (LeCroy binary format). For additional information on files format, contact your nearest LeCroy representative.

Save Waveform (Binary)
The Save Waveform command will store your active waveform in your PXA125 directory, as a binary file with an *.wav extension. If this is the first time you save your waveform, the Save Waveform As… command will be invoked automatically, letting you select name, location and format for your waveform file.

Save Waveform As…
Use the Save Waveform As… command the first time you save your waveform. It will let you select name, location and format for your waveform file. The Save Wave As… dialog box is shown in Figure 3-15.



Save in: PXA125

File name: Sample

Save as type: Binary (*.wav)

Figure 3-45. The Save Wave AS… Dialog Box

Edit Commands

The Edit commands are used for manipulating the waveform that is drawn on the screen. The editing commands are explained in the following paragraphs.

Autoline

The Autoline command lets you draw straight-line segments. To draw a line using this command click the left mouse button at the start point. Click again at the next point and then click on the right mouse button to terminate this operation.

Sketch

The Sketch command lets you draw free-hand segments. To draw a line using this command click and hold the left mouse button at the start point. Release the mouse button when you want to stop and then click on the right mouse button to terminate this operation.

Smooth

The Smooth command lets you smooth out rough transitions on your waveform. This is done mathematically by multiplying waveform coordinates by the non-linear portion of a cubic parabola.

The Smooth operation is done on segments of the waveform that are bound by anchors. Anchor operation is described later in this chapter. Place you anchors on the left and right of your waveform segment and select the Smoot command. The waveform will change its shape immediately to follow the mathematical pattern of a parabolic curve.

Note that small segments with fast transitions, when combined with parabolic expressions have tendencies to generate even larger transitions. Therefore, make sure you omit such sections of the waveform when you use this operation.

Filter

The Filter used with this command is moving average. This is done by recalculating each point as an average of symmetrical number of adjacent points. When you select the Filter command, a dialog box pops up, letting you program the filter spacing in number of adjacent points. You can filter the entire waveform, or you may chose to filter a segment of the waveform by placing the anchors as boundaries on the left and right of the segment.

Invert

The Invert command lets you invert the entire waveforms, or marked segments of waveforms. The waveform is inverted about the 0-point axis.

Trim Left

The Trim Left command operates in conjunction with the left anchor. It is helpful if you want to “chop off “ the left section of your waveform. After you use this command, the portion to the left of the anchor will be removed from the screen and the waveform will start from the point where the anchor was placed before the trim left operation. When using this command, bear in mind that the number of points that you use for your waveform must divide by 4. If you placed your anchor in between 4-point increments then the waveform length will automatically adjust to the nearest legal size.

Trim Right

The Trim Right command operates exactly as the Trim Left command except this command affects the portion to the right of the right anchor.

Unmark

The unmark command removes the anchors from the waveform screen and resets anchor positions to point 0 and the last waveform point.

Undo

The Undo command undoes the last editing operation.

View Commands

The View commands have commands that let you view various options on the waveform area. The View commands include: View All, Channel 1 Waveform, Return to Panel and Hide/Show Toolbar. Description of the view commands is given in the following.

Zoom In

You may zoom in on portions of the waveform by dragging the left and right anchors to the area you want to zoom, then use the Zoom In command. The example in Figure 3-16 shows a magnified portion of the waveform while the smaller window shows the complete waveform map and the portion you magnified. You can click and drag the white section to pan the magnified portion. While zoomed in you can perform autoline and sketch editing, or zoom-in further by dragging the left and right anchors then using the Zoon In command again.

Zoom Out

The Zoom out command restores the waveform to its original size and removes the waveform map window from the screen.

View Toolbars

The View Toolbars command lets you hide or show the various toolbars that are available for the Wave Composer. The available toolbars are: Link bar, File bar, Wave Edit bar and System bar.

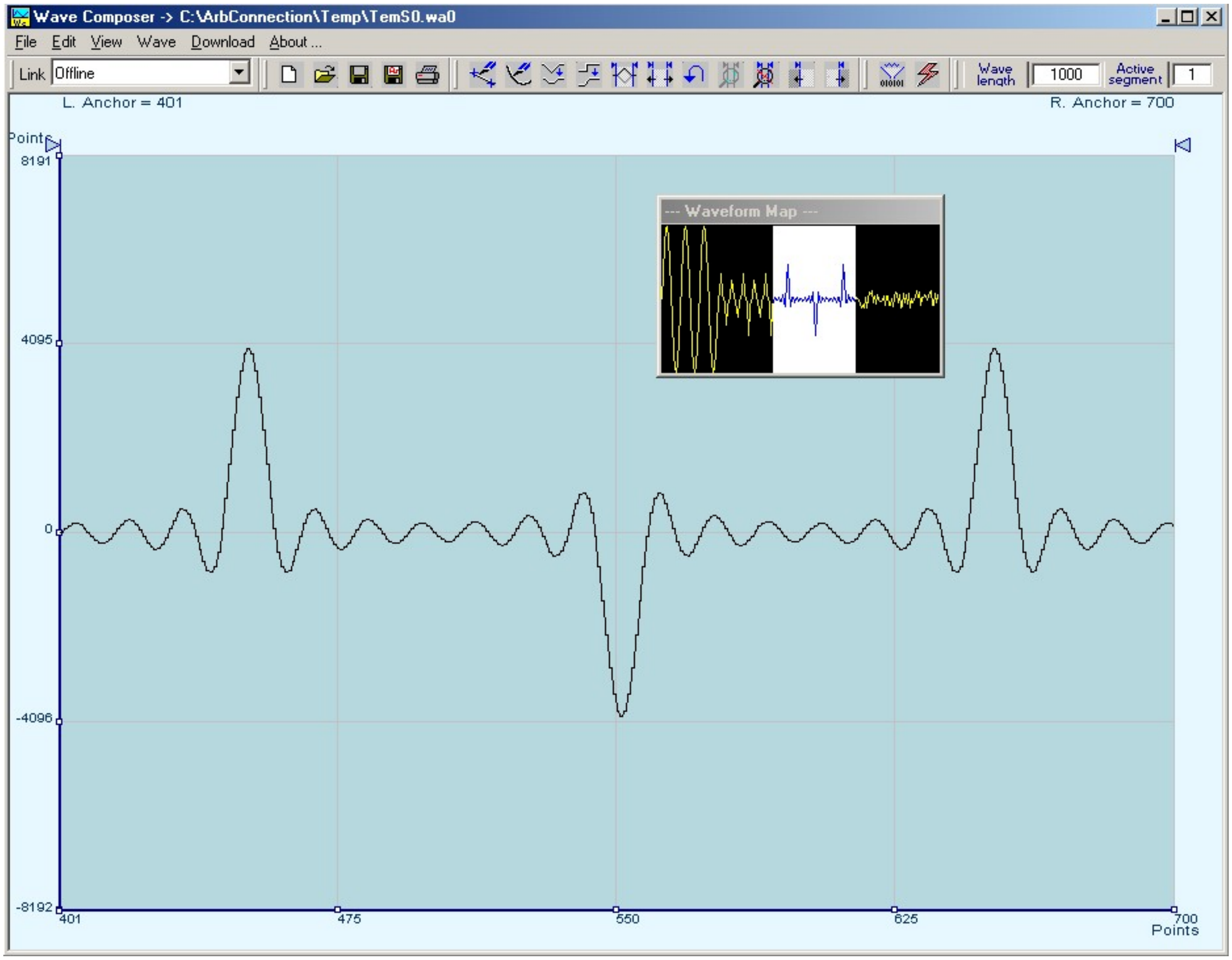
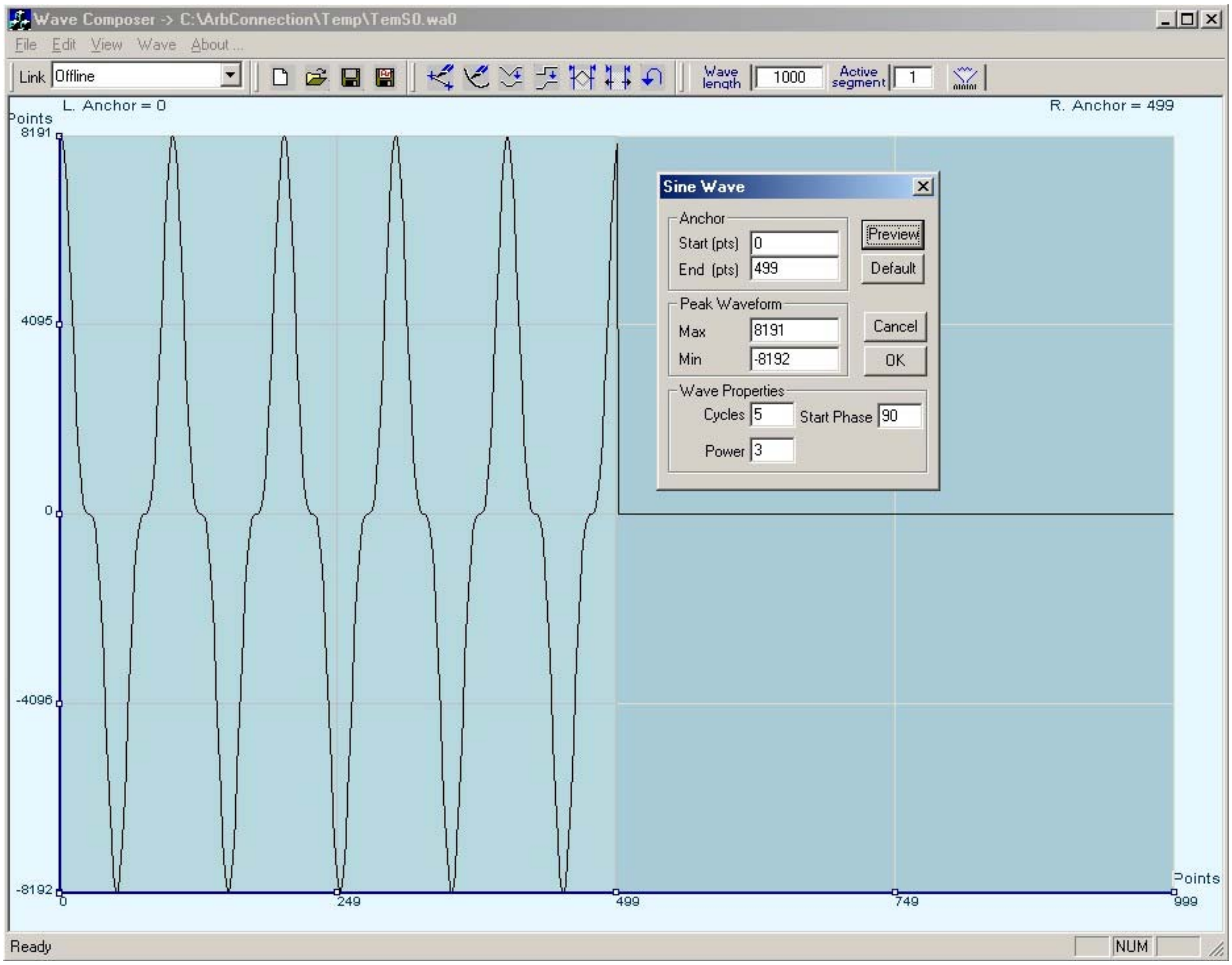

Wave Composer -> C:\ArbConnection\Temp\TemS0.wa0
File Edit View Wave Download About ...
Link Offline
Wave length 1000 Active segment 1
L. Anchor = 401
R. Anchor = 700
Points
8191
4095
0
-4096
-8192
401
475
550
625
700
Points
--- Waveform Map ---

Figure 3-46. Zooming In on Waveforms

Wave Commands The Wave commands let you create waveforms on the screen. The Wave command has a library of 8 waveforms: Sine, Sawtooth, Square, Sinc, Gaussian, Exponent, Pulse, and Noise. It also lets you create waveforms using the Equation Editor. Information how to create waveforms using the Wave commands is given below.

Creating Waveforms From the Built-in Library

You can create any waveform from the built-in library using the Wave command. Clicking on one of the Wave options will open a dialog box. An example of the Sine waveform dialog box is shown in Figure 3-17. This dialog box is representative of the rest of the waveforms, so other waveforms will not be described.



Link: Offline
Wave length: 1000
Active segment: 1
L. Anchor = 0
R. Anchor = 499

Anchor
Start (pts): 0
End (pts): 499

Peak Waveform
Max: 8191
Min: -8192

Wave Properties
Cycles: 5
Start Phase: 90
Power: 3

Figure 3-47. An Example of Generating Sine Waveforms From the Built-in Library.

Creating Sine Waveforms

Use the following procedure to create sine waveforms from the built-in library. Click on Wave, then sine… the dialog box as shown in Figure 3-17 will appear. You can now start programming parameters that are available in this box.

Start Point – Defines the first point where the created wave will start. Note that if you change the start point the left anchor will automatically adjust itself to the selected start point. The example shows start point set at point 0.

End Point – Defines where the created waveform will end. Note that as you change the end point the right anchor will automatically adjust itself to the selected end point. The example shows end point set at point 499.

Cycles – The Cycles parameter defines how many sine cycles will be created within the specified start and end points. The example below shows five sine cycles.

Waveform Amplitude – 14-bit of vertical define 16,384 incremental steps. The Amplitude parameter defines how many of these steps are used for generating the sine. The waveform amplitude is given in units of points. The waveform amplitude fields let you design the maximum and minimum peak separately. The example is showing sine waveform with maximum peak-to-peak values. Smaller values will generate an attenuated sine.

Start Phase – The start phase parameter defines the angle of which the sine will start. The example shows 90° start phase.

Power – The example shows sine cubed. Sine to the power of 1 will generate a perfect sine. Power range is from 1 through 9.

System Commands

The System commands let you do two actions: Delete all segments and download the active waveform to the selected channel. Both of these commands are also available as icons on the toolbar.

The Toolbars

The toolbar contains icons for editing the waveform screen, icons for saving and loading waveforms, fields for selecting an active segment and for adjusting segment length and more. The Toolbar is shown in Figure 3-18 and the individual icons are described in the following.

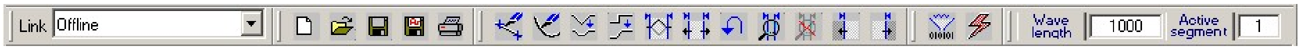


Figure 3-48. The Toolbar Icons

This icon represents the Open New Waveform file command. When you click on this icon, the current waveform will be removed from the waveform area and a DC line at the 0 coordinate will be displayed.

This icon represents the Open Waveform file command. When you click on this icon, the Open Waveform dialog box will open and will let you select a waveform file to display on the waveform screen.

This icon will save the file you currently have on your waveform screen. The file will be saves in *.wav format that is unique to the wave composer.

The Save As… icon lets you save your waveform as a file. The file will be saves in *.wav format that is unique to the wave composer. It also can be saved as *.asc in ASCII format.

Use this icon to print your waveform screen on your default printer. With this command you may print the active Waveform Window. The standard printer dialog box will appear and will let you select printer setup, or print the waveform page.

Click on this icon to start the autoline operation. You can exit the autoline mode by clicking on this icon again, or by clicking on the right mouse button. The Autoline command lets you draw straight-line segments. To draw a line using this command click the left mouse button at the start point. Click again at the next point and then click on the right mouse button to terminate this operation.

The sketch mode is invoked using this icon. Click on this icon again, or click your right mouse button to terminate the sketch mode. The Sketch command lets you draw free-hand segments. To draw a line using this command click and hold the left mouse button at the start point. Release the mouse button when you want to stop and then click on the right mouse button to terminate this operation.

Clicking on this icon will invoke the smooth function. The Smooth command lets you smooth out rough transitions on your waveform. This is done mathematically by multiplying waveform coordinates by the non-linear portion of a cubic parabola.

The Smooth operation is done on segments of the waveform that are bound by anchors. Anchor operation is described later in this chapter. Place you anchors on the left and right of your waveform segment and select the Smoot command. The waveform will change its shape immediately to follow the mathematical pattern of a parabolic curve.

Note that small segments with fast transitions, when combined with parabolic expressions have tendencies to generate even larger transitions. Therefore, make sure you omit such sections of the waveform when you use this operation.

This icon will filter operation. The Filter used with this command is moving average. This is done by recalculating each point as an average of symmetrical number of adjacent points. When you select the Filter command, a dialog box pops up, letting you program the filter spacing in number of adjacent points. You can filter the entire waveform, or you may chose to filter a segment of the waveform by placing the anchors as boundaries on the left and right of the segment.

An easy way of inverting your waveform is by using the invert function. The Invert command lets you invert the entire waveforms, or marked segments of waveforms. The waveform is inverted about the 0-point axis.

Click on this icon to removes the anchors from the waveform screen and resets anchor positions to point 0 and the last waveform point.

Use this icon to undo your last editing operation. Only one step will back up.

Use this icon to zoom in on an area enclosed between one or two anchors. To move the anchors click and hold on the triangle on top of the waveform screen, than drag the anchor to the desired location. If you want to specify anchor location, place your mouse on the triangle and click the right button on your mouse type in the anchor location and press Enter.

Use this icon after you have zoomed in to view the entire waveform area.

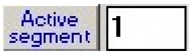
When you use this icon, the waveform to the left of the left anchor will be removed and the anchor position will become the first point of your waveform. Note that the length of your horizontal axis must divide by four therefore, if you place the left anchor at a position other than a multiple of four, you’ll be asked to adjust the size accordingly.

When you use this icon, the waveform to the right of the right anchor will be removed and the anchor position will become the last point of your waveform. Note that the length of your horizontal axis must divide by four therefore, if you place the right anchor at a position other than a multiple of four, you’ll be asked to adjust the size accordingly.

Use extreme caution when using this icon because it will delete all of your segments from your waveform memory.

When you click on this button, the waveform that is displayed on your waveform screen will be downloaded to the active segment on the selected channel.

The wavelength field lets you program the length of the active segment. You can program up to 4096 individual segments that each has a unique length. Minimum segment length is 16 points and maximum is 4,194,303 ("0" is counted as a point for a total of 4,194,304). To program a new length for the active segment click on the edit field, use your backspace or delete keys to clear the field and type in the new number. To complete the selection, click on the Wavelength button.



Use this edit field to define the active segment you intend to program. The PXA125 lets you program up to 4096 segments, of which each could be programmed with a unique waveform length. To select a new active segment click on the edit field, use your backspace or delete keys to clear the field and type in the new number. To complete the selection, click on the Active Segment button.

The Waveform Screen

Waveforms are created and edited on the waveform screen. Figure 3-19 shows an example of a waveform created using the equation editor and the anchors to limit generation of the waveform between points 100 and 900. The various elements of the waveform screen are described below.

The waveform screen has two axes – vertical and horizontal. Both axes are divided into points.

The vertical axis is labeled from –8192 through 8191 for a total of 16,384 point. This number represents 14 bits of vertical resolution and cannot be changed because it is critical to the range of which the PXA125 operates.

The horizontal axis, by default has 1000 points (from point 0 to 999). This number can be changed using the Wave Length field in the Toolbar. The maximum length depends on the option installed in your instrument. The wave composer will let you define the horizontal axis to a maximum of 4,194,304 (4 Meg words).

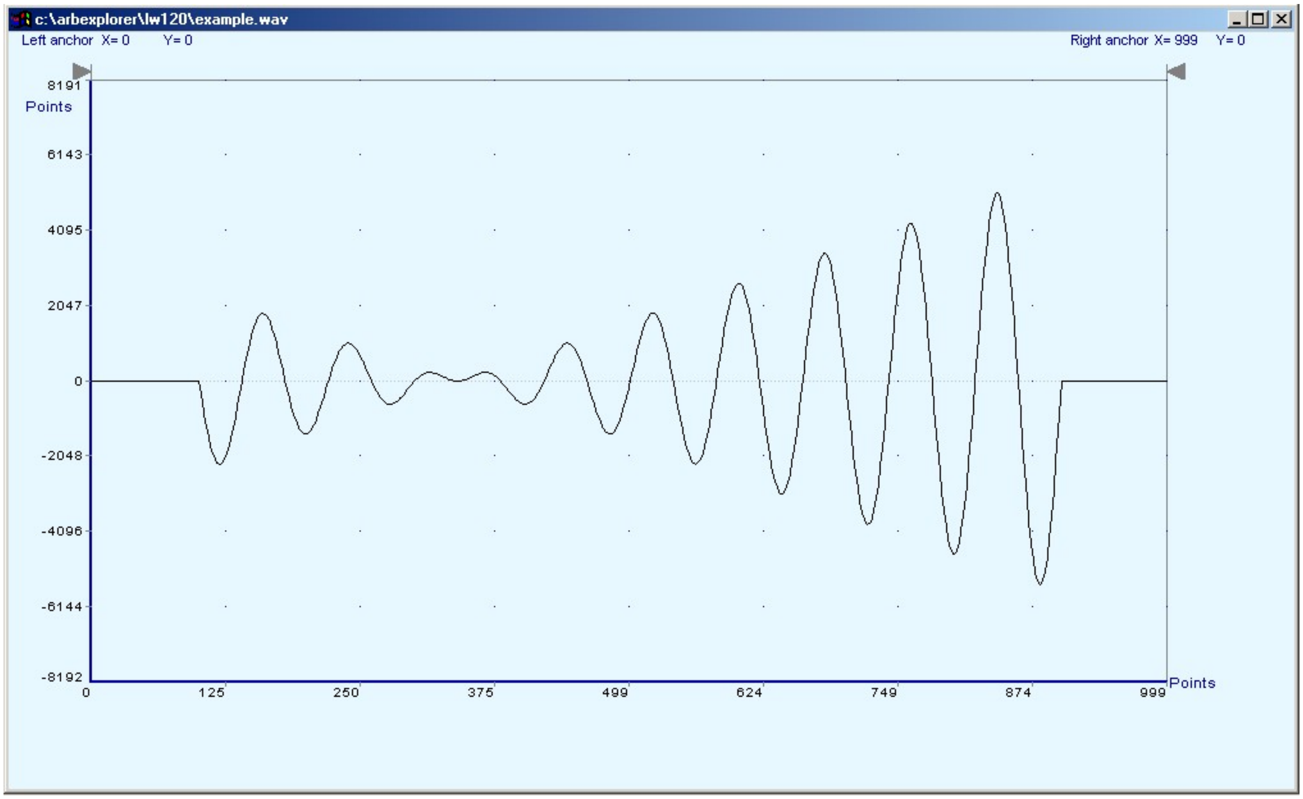


Figure 3-49. The Waveform Screen

Notice on the left top and on the right top there are two triangles pointing to the center of the screen. These are the anchors. The anchors are used as the start and end pointers where your waveform will be created. For example, if you want to create a sine waveform between point 100 and point 500, you place the left anchor at point 100 and the right at point 500 and then generate the sine from the built-in library.

There are two ways to control anchor placements.

1) Click and hold your mouse cursor on the left anchor triangle and then drag the curtain to the left position. Do the same for the right anchor. Notice the X and Y coordinates at the top of the waveform screen and how they change to correspond to your anchor placement.

2) You can also place your anchors in a more precise manner from the waveform library by programming the start and end points for the waveform. An example of anchor placement using the sine dialog box is shown in Figure 3-17.

Finally, when you are done creating and editing your waveform, you can save your work to a directory of your choice. The name at the title will show you the name you selected for storing your waveform and its path.

The FM Composer

The FM Composer looks and feels almost like the waveform composer except there is a major difference in what it does. If you look at the opening screen as shown in Figure 3-20, you'll see that the vertical axis is marked with sample clock frequencies. You'll see later that as you draw waveforms on the FM composer screen, these waveforms will represent frequency changes and not amplitude changes as are generated by the waveform composer.

The FM composer is a great tool for controlling frequency agility by generating the agility curve as an arbitrary waveform. For example, if you create a sine waveform, the PXA125 will generate frequency-modulated signal that will follow the sine pattern. The resolution and accuracy of the modulated waveform is unsurpassed and can only be duplicated by mathematical simulation. The FM composer is loaded with many features and options so use the following paragraphs to learn how to create and download modulating waveforms to the PXA125 using the FM Composer.

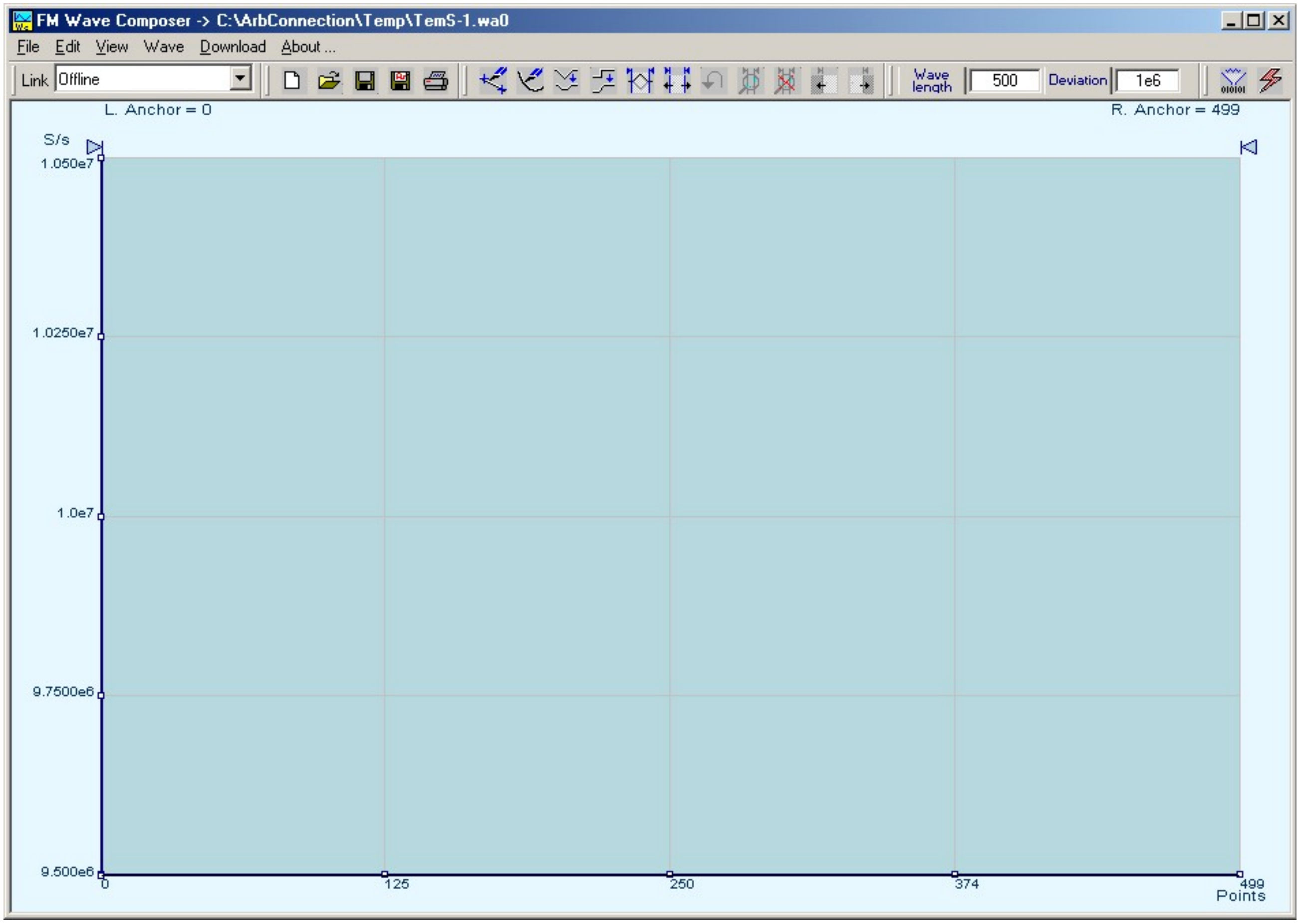
You can invoke the FM Composer from two places: On the Panels bar click on FM, or from the Modulation Panel click on “To FM Composer”. Figure 3-20 shows the waveform composer, immediately after you first invoke this utility.

The Wave Composer has three sections: Commands bar, Toolbar and Waveform screen. Refer to Figure 3-20 throughout the description of these parts.

The Commands bar

The commands bar provides access to standard Windows commands such as File and View.

In general, clicking on one of the commands opens a dialog box with an additional list of commands. Then, clicking on an additional command, may open a dialog box, or generate an immediate action. For example, Clicking on File and then Exit will cause an immediate termination of the FM Composer. On the other hand, clicking on Wave and then on Square, will open a Square Wave dialog box that lets you program and edit square wave parameters. The various commands in the Commands bar are listed and described below.



FM Wave Composer -> C:\ArbConnection\Temp\TemS-1.wa0

Link Offline

Wave length 500

Deviation 1e6

L. Anchor = 0

R. Anchor = 499

Figure 3-50. The FM Composer Opening Screen

File Commands The File command has 4 command lines that control waveform files. Also use this command to exit the FM composer program. Description of the various commands under File is given below.

New Waveform

The New Waveform command will remove the waveform from the screen. If you made changes to the waveform area and use this command, you should save your work before clearing the screen. The New Waveform command is destructive to the displayed waveform.

Open Waveform…

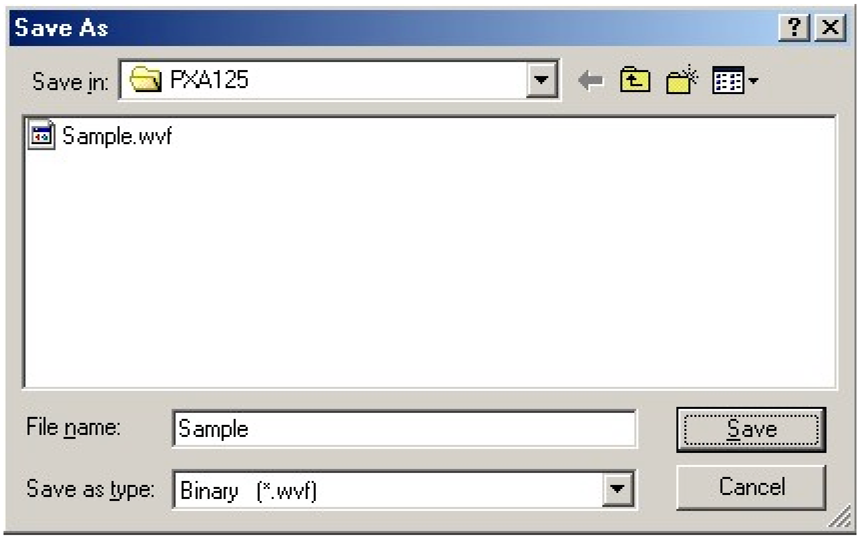
The Open Waveform… command will let you brows your disk for previously saved waveform files and load these waveforms to the waveform area. This command is also very useful for converting waveform files to format that is acceptable by the Wave Composer. The Open Waveform command can convert ASCII. *ASF (space delimited text). For additional information on files format, contact your nearest LeCroy representative.

Save Waveform

The Save Waveform command will store your active waveform in your PXA125 directory, as a binary file with a *.wvf extension. If this is the first time you save your waveform, the Save Waveform As… command will be invoked automatically, letting you select name, location and format for your waveform file.

Save Waveform As…

Use the Save Waveform As… command the first time you save your waveform. It will let you select name, location and format for your waveform file. The Save As… dialog box is shown in Figure 3-21.



Save As

Save in: PXA125

Sample.wvf

File name: Sample

Save as type: Binary (*.wvf)

Figure 3-51. The FM Composer Save AS… Dialog Box

Exit

The Exit command ends the current FM Composer session and takes you back to the Panels screen. If you made changes to your waveform since it was last saved, make sure to Save your work before you use this command.

Edit Commands

The Edit commands are used for manipulating the waveform that is drawn on the screen. The editing commands are explained in the following paragraphs.

Autoline

The Autoline command lets you draw straight-line segments. To draw a line using this command click the left mouse button at the start point. Click again at the next point and then click on the right mouse button to terminate this operation.

Sketch

The Sketch command lets you draw free-hand segments. To draw a line using this command click and hold the left mouse button at the start point. Release the mouse button when you want to stop and then click on the right mouse button to terminate this operation.

Smooth

The Smooth command lets you smooth out rough transitions on your waveform. This is done mathematically by multiplying waveform coordinates by the non-linear portion of a cubic parabola.

The Smooth operation is done on segments of the waveform that are bound by anchors. Anchor operation is described later in this chapter. Place you anchors on the left and right of your waveform segment and select the Smoot command. The waveform will change its shape immediately to follow the mathematical pattern of a parabolic curve.

Note that small segments with fast transitions, when combined with parabolic expressions have tendencies to generate even larger transitions. Therefore, make sure you omit such sections of the waveform when you use this operation.

Filter

The Filter used with this command is moving average. This is done by recalculating each point as an average of symmetrical number of adjacent points. When you select the Filter command, a dialog box pops up, letting you program the filter spacing in number of adjacent points. You can filter the entire waveform, or you may chose to filter a segment of the waveform by placing the anchors as boundaries on the left and right of the segment.

Invert

The Invert command lets you invert the entire waveforms, or marked segments of waveforms. The waveform is inverted about the 0-point axis.

Unmark

The unmark command removes the anchors from the waveform screen and resets anchor positions to point 0 and the last waveform point.

Undo

The Undo command undoes the last editing operation.

Wave Commands

The Wave commands let you create waveforms on the screen. The Wave command has a library of 6 waveforms: Sine, Triangle, Square, Exponent, Pulse, and Noise. It also lets you create waveforms using an Equation editor. Information how to create waveforms using the Wave commands is given below.

Creating Waveforms From the Built-in Library

You can create any waveform from the built-in library using the Wave command. Clicking on one of the Wave options will open a dialog box. An example of the Sine waveform dialog box is shown in Figure 3-22. This dialog box is representative of the rest of the waveforms, so other waveforms will not be described.

Creating Sine Waveforms

Use the following procedure to create sine waveforms from the built-in library. Click on Wave, then sine… the dialog box as shown in Figure 3-22 will appear. You can now start programming parameters that are available in this box.

Start Point Anchor – Defines the first point where the created wave will start. Note that if you change the start point the left anchor will automatically adjust itself to the selected start point. The example shows start point set at point 200.

End Point Anchor – Defines where the created waveform will end. Note that as you change the end point the right anchor will automatically adjust itself to the selected end point. The example shows end point set at point 499.

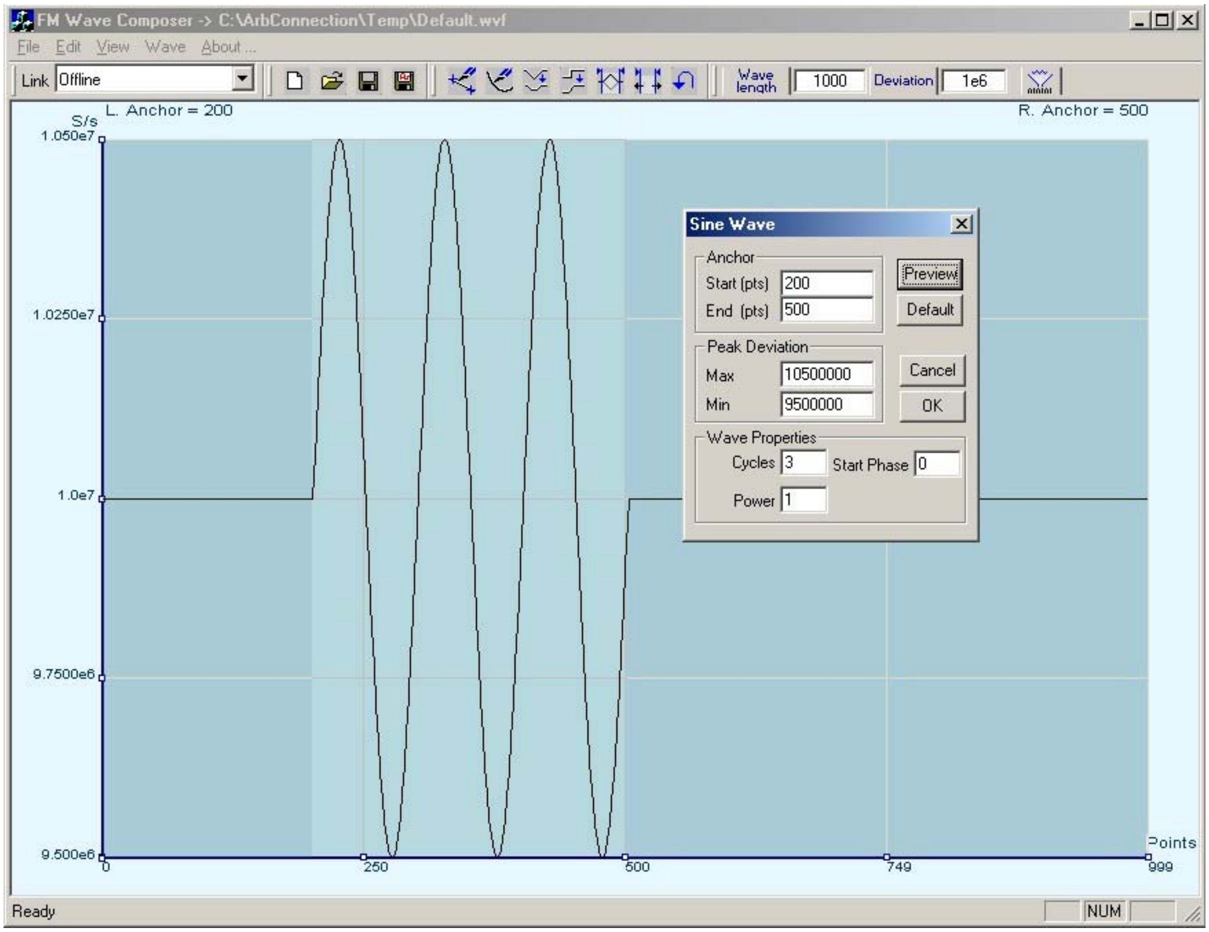
Max. **Peak Deviation** – This parameter defines the forward peak deviation. Note that the forward peak deviation cannot exceed the pre-defined Deviation parameter as shown on the Toolbar. In case you need to exceed the pre-defined peak value you must quit this box and modify the Deviation parameter to provide sufficient range for the forward peak deviation range.

Min. **Peak Deviation** – This parameter defines the backwards peak deviation. Note that the backwards peak deviation cannot exceed the pre-defined Deviation parameter as shown on the Toolbar. In case you need to exceed the pre-defined peak value you must quit this box and modify the Deviation parameter to provide sufficient range for the backwards peak deviation range.

Cycles – The Cycles parameter defines how many sine cycles will be created within the specified start and end anchor points. The example below shows three sine cycles.

Start Phase – The start phase parameter defines the angle of which the sine will start. The example shows 0° start phase.

Power – Sine to the power of 1 will generate a perfect sine. Power range is from 1 through 9.



Link: Offline

Wave length: 1000

Deviation: 1e6

L. Anchor = 200

R. Anchor = 500

Sine Wave

Anchor

Start (pts): 200

End (pts): 500

Peak Deviation

Max: 10500000

Min: 9500000

Wave Properties

Cycles: 3

Start Phase: 0

Power: 1

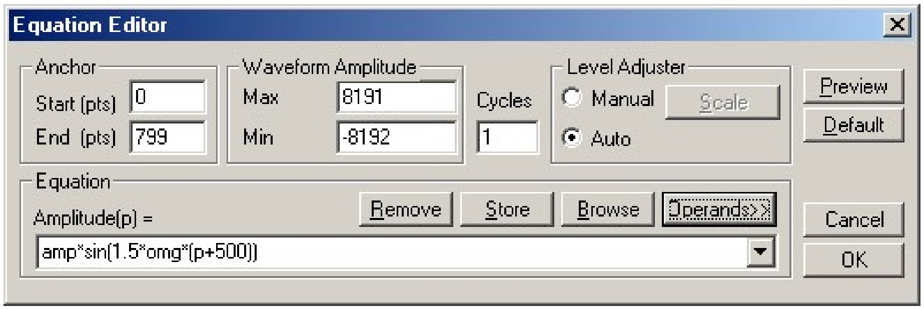
Figure 3-52. An Example of Generating Modulating Sine From the Built-in Library.

Generating Waveforms Using the Equation Editor

One of the most powerful feature within ArbConnection and probably the feature that will be used most is the Equation Editor. The Equation Editor let you write equations the same way as you would do on a blank piece of paper. The equations are then translated to sequential points that form waveforms and are displayed on the waveform screen. The Equation Editor will detect and inform you on syntax errors and, with its self adjusting feature, will automatically adjust your parameters so that none of the points on your waveform will exceed the maximum scale limits.

When you invoke the Equation Editor, the dialog box, as shown in Figure 3-23 will display. Use the following paragraphs to learn how to use this dialog box and how to write your equations.

There are three main sections in the equation editor plus control buttons. These sections are described below.



Equation Editor

Anchor
Start (pts) 0
End (pts) 799

Waveform Amplitude
Max 8191
Min -8192

Cycles 1

Level Adjuster
○ Manual
● Auto

Equation
Amplitude(p) = amp*sin(1.5*omg*(p+500))

Figure 3-53. The Equation Editor Dialog Box

Anchor The Anchor section has two fields: Start and End Point.

Start (pts) – Defines the first point where the created wave will start. Note that if you change the start point the left anchor will automatically adjust itself to the selected start point.

End (pts) – Defines where the created waveform will end. Note that as you change the end point the right anchor will automatically adjust itself to the selected end point.

Level Adjuster The Level Adjuster is a convenient tool that helps you adjust the amplitude and offset without modifying your equation. The Level Adjuster mode does not interfere with your calculations and displays the waveform as computed from your equation. The only difference is that your final calculations are stretched or shrunk or offset on the vertical scale to fit the new amplitude and offset boundaries.

For example, look at the equation that is shown in Figure 3-23. This equation will generate a waveform similar to what you can see on Figure 3-22. If you change the offset setting in the Offset field and press the Adjust key, your waveform will offset immediately without changing the equation. The same way, you can also change the amplitude, or both amplitude and offset. If you check the Manual option, you’ll have to click on the Adjust button for the Amplitude and Offset parameters to take effect. The Adjust button name will change to Restore and back to Adjust if you click on it again. If you check the Auto option, your waveform will be created automatically with the new

Amplitude and Offset setting.

Equation You will be using the Equation field for writing your equations. Equation syntax and conventions are discussed in the following.

Control Buttons There are five control buttons at the right corner of the dialog box. Use the Preview button to preview an image of your equation, or use the OK button to place your waveform on the waveform screen and to leave the dialog box. The Browse button will open the waveform browser so you can search for pre-stored waveforms to add to your equation. The Operands button will expand the dialog box to show all of the operands that are available for use in the Equation Editor and the Cancel button will remove the dialog box from the screen and will discard of any waveforms that you previewed with your Equation Editor.

Writing Equations

The Equation Editor lets you process mathematical expressions and convert them into waveform coordinates. As you probably already know, waveforms are made of vertical samples. The number of samples on your waveform is determined by the wavelength parameter. For example, if you have 1024 horizontal points, your equation will be computed along 1024 points as a function of the vertical scale. Each vertical sample is computed separately and placed along the horizontal axis. The points are graphically connected to form a continuous waveform. However, if you zoom in on a waveform line, you'll see that the points are connected like a staircase. In reality, the PXA125 generates its waveforms exactly as shown on the screen however, if the waveform has many horizontal points, the steps are hard to see.

Equations are always computed as a function of the vertical (Amplitude) axis therefore the left side of your equation will always look as Ampl(p)=, where “p” is the equation variables in units of waveform points. You can write equations with up to 256 characters. If the equation is too long to fit in the visible field, parts to the left or right will scroll off the ends.

Equation Conventions

The following paragraphs describe the conventions that are used for writing an equation. To avoid errors, it is extremely important that you make yourself familiar with these conventions before you plan your waveforms.

Equations are written in conventional mathematical notation. You may only enter the right part of the equation. The only limitation is that the equation must be of a single variable that is directly related to the current horizontal axis setting. Case is not important and spaces are ignored. Numbers are entered in scientific notation. All calculations are done with double-digit precision. For the trigonometric functions, all angles are expressed in radians.

A number of constants are provided: e, which is the base of the natural logarithm; pi, which is the circumference of a unit-diameter circle; per, which equals the programmed horizontal range; f, which equals 1 /per; omg, which equals 2 * pi / per, and numerals from -1E^20 <> 1E^20.

There are three classes of precedence: ^ (raise to power) has the highest precedence; (multiply) and / (divide) come second; + and - have the lowest precedence. Parentheses may be used to change the order of precedence. The following table summarize the mathematical expressions and their respective abbreviated commands that can be used with the Equation Editor.

Equation Editor Operands

^	Raise to the power
*	Multiply
/	Divide
+	Add
-	Subtract
( )	Parentheses
e	Base of natural Logarithm
pi (π)	Circumference of unit-diameter circle
per	Horizontal wavelength in points
f	I/per
omg (Ω)	2*π / per
amp	Amplitude in units of points or seconds
sin(x)	The sine of x(*)
cos(x)	The cosine of x
tan(x)	The tangent of x
ctn(x)	The cotangent of x
log(x)	The base IO logarithm of x
In(x)	The natural (base e) logarithm of x
abs(x)	The absolute value of x
-1E^20<>1E^20	Numerals, equation constants

(*)x = argument mathematical expression

After you got yourself familiar with the operands and conventions, you can commence with a few simple equations and see what they do your waveform screen. Once you'll get the feel, you'll be able to explore your own creativity to generate much more complicated and complex waveforms.

Typing Equations

If you remember from your old high school studies, the simplest curve of Y as a function of X is defined by the equation Y=aX+b. You can use the same “technique” to generate straight lines with the Equation

Editor. Assuming first that p=0, try this:

Ampl(p)=1000

Press [Preview] and see what you get. Of course, you get an uninteresting line that runs parallel to the X-axis. Now, lets give the line some angle by typing:

Ampl(p)=-2*p+2000

Press [Preview] and see that the line slopes down. It may still be not very interesting however, pay close attention to the convention that is used in this equation. You cannot type: Ampl(p)=-2p+1000, like you would normally do in your notebook; You must use the * (multiply) sign, otherwise you'll get a syntax error. Now we'll try to generate a simple sine waveform. Try this:

Ampl(p)=sin(10)

Press [Preview] and… sorry, you still get nothing on the screen. The Wave Composer did not make a mistake! The sine of 10 in radians is exactly what it shows. You are unable to see the result because the line on your screen running across the 0 vertical point.

REMEMBER

The equation must be a function of a single variable and that variable must be directly related to the Horizontal axis Scale setting.

Now try this:

Ampl(p)=sin(omg*p)

Still no good, but now press the [Adjust] button and here is your sinewave. So what's wrong? Well, if you'll give it a little amplitude it might help so, do it now exactly as follows:

Ampl(p)=6000*sin(omg*p)

There you go. You should now see a perfect sine waveform with a period of 1000 points. This is because you have asked the Equation Editor to compute .the sine along p points (“p” is the equation variable, remember?). If you want to create 10 sine waveforms, you should multiply p by 10. Try this:

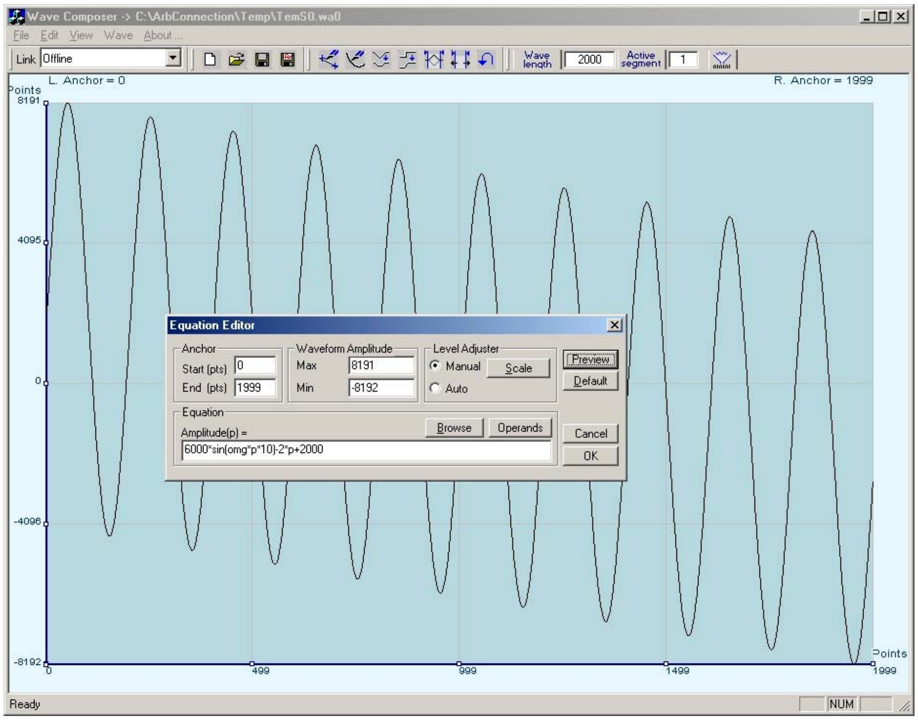
Ampl(p)=6000*sin(omg*p*10)

Equation Samples

So far, you have learned how to create two simple waveforms: straight lines and trigonometric functions. Lets see if we can combine these waveforms to something more interesting. Take the straight line equation and add it to the sinewave equation:

Ampl(p)=6000*sin(omg*p*I0)-2*p+2000

Press [Preview]. Your screen should look like Figure 3-24.



Wave Composer -> C:\ArbConnection\Temp\TemS0.wa0

Link: Offline

Wave length: 2000

Active segment: 1

L. Anchor = 0

R. Anchor = 1999

Equation Editor

Anchor

Start (pts): 0

End (pts): 1999

Waveform Amplitude

Max: 8191

Min: -8192

Level Adjuster

● Manual

○ Auto

Equation

Amplitude(p) = 6000*sin(omg*p*10)-2*p+2000

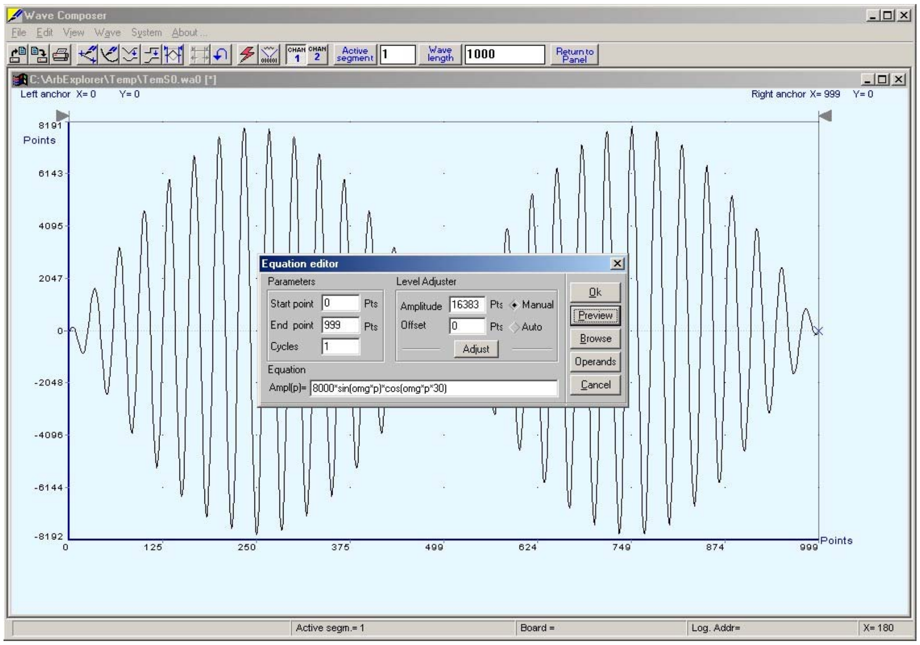
Ready

Figure 3-54. Using the Equation Sample to Combine Two Waveforms

Now lets try to modulate two sinewaves with different periods and different start phase. Type this:

Ampl(p)= 8000*sin(omg*p)*cos(omg*p*30)

Press [Preview]. Your screen should look like Figure 3-25.



Active segment: 1

Wave length: 1000

Start point: 0 Pts

End point: 999 Pts

Cycles: 1

Amplitude: 16383 Pts ● Manual

Offset: 0 Pts ○ Auto

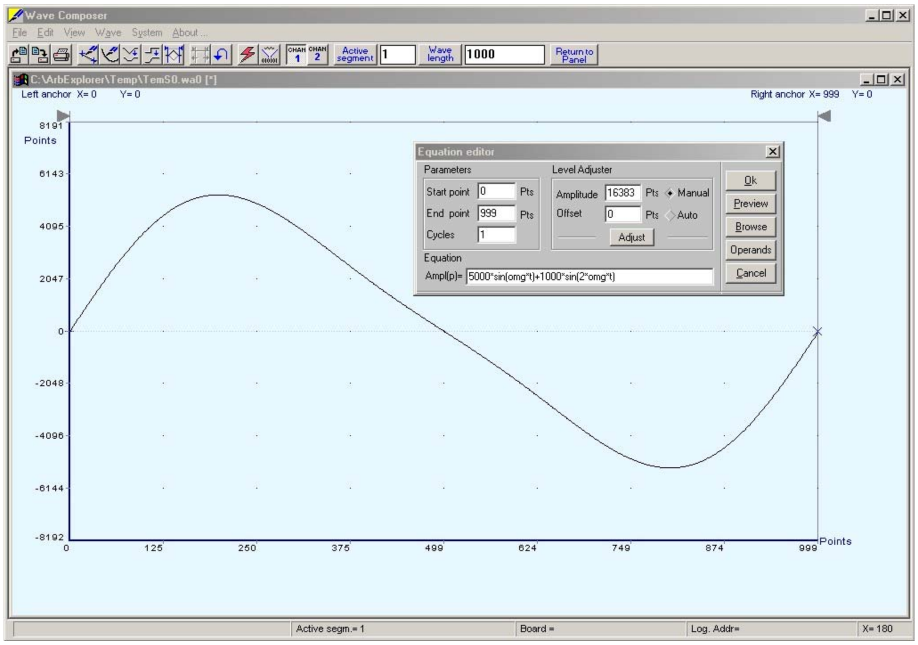
Ampl(p)= 8000*sin(omg*p)*cos(omg*p*30)

Figure 3-55. Using the Equation Editor to Modulate Sine Waveforms.

In the following example, as shown in Figure 3-23, 20% second harmonic distortion has been added to a standard sinewave. The original waveform had a peak-to-peak value of 5000 points so 20% second harmonic is equivalent to 1000 points. The frequency of the second harmonic is obviously double that of the fundamental, so term +1000*sin(2*omg*t) is added to the original sine wave equation. Use the following equation:

Ampl(p)=5000*sin(omg*t)+1000*sine(2*omg*t)

Press [Preview]. Your screen should look like Figure 3-26.



Active segment: 1

Wave length: 1000

Left anchor X= 0 Y= 0

Right anchor X= 999 Y= 0

Start point: 0 Pts

End point: 999 Pts

Cycles: 1

Amplitude: 16383 Pts ● Manual

Offset: 0 Pts ○ Auto

Equation

Ampl(p)= 5000*sin(omg*t)+1000*sin(2*omg*t)

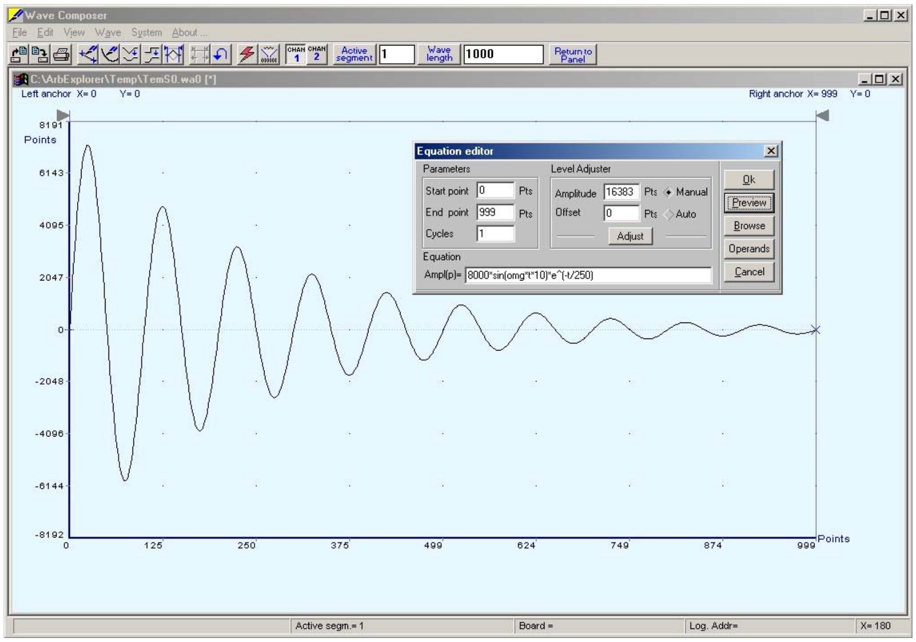
Figure 3-56. Using the Equation Editor to Add Second Harmonic Distortion.

In Figure 3-24 we created 10 cycles of sinewave made to decay exponentially. The original expression for a standard sinewave is multiplied by the term e^(-t/250). Increasing the value of the divisor (200 in this case) will slow down the rate of decay.

Use the following equation:

Ampl(p)=8000*sin(omg*t*10)^e(-t/250)

Press [Preview]. Your screen should look like Figure 3-27.



Active segment: 1

Wave length: 1000

Start point: 0 Pts

End point: 999 Pts

Cycles: 1

Amplitude: 16383 Pts ● Manual ○ Auto

Offset: 0 Pts

Equation

Ampl(p)= 8000*sin(omg*t*10)*e^(-t/250)

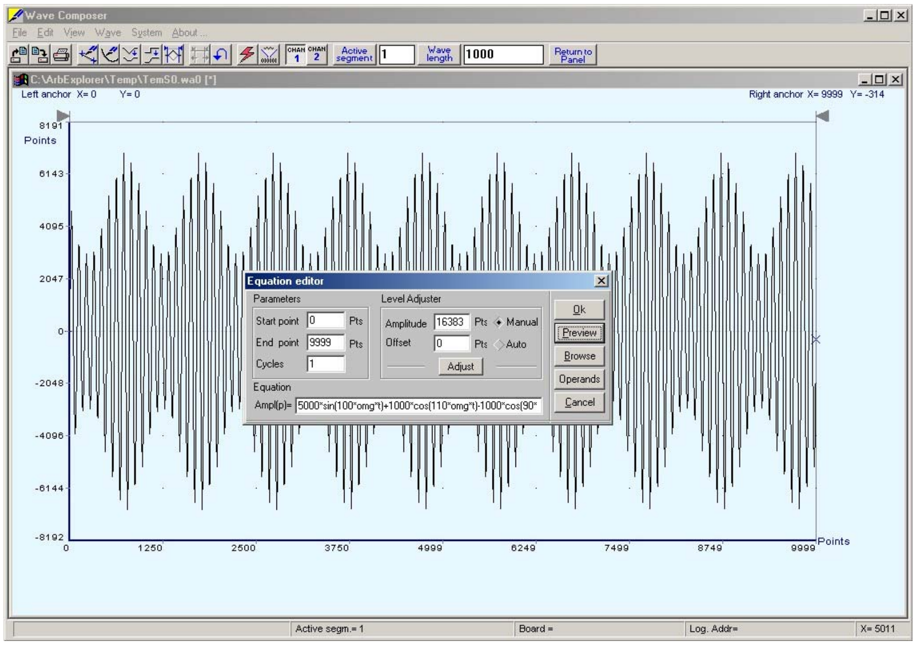
Figure 3-57. Using the Equation Editor to Generate Exponentially Decaying Sinewave

The last example as shown in Figure 3-25 is the most complex to be discussed here. Here, 100 cycles of sinewave are amplitude modulated with 10 cycles of sine wave with a modulation depth of 20%. To achieve this, the upper and lower sidebands are defined separately and added to the fundamental or carrier. The upper sideband is produced by the expression 100*cos(110*omg*t) and the lower sideband by the term 100*cos(90*omg*t).

Use the following equation:

Ampl(p)=5000*sin(100*omg*t)+1000*cos(110*omg*t)-1000*cos(90*omg*t)

Press [Preview]. Your screen should look like Figure 3-28.



Active segment: 1

Wave length: 1000

Start point: 0 Pts

End point: 9999 Pts

Cycles: 1

Amplitude: 16383 Pts ● Manual

Offset: 0 Pts ○ Auto

Equation

Ampl(p)= 5000*sin(100*omg*t)+1000*cos(110*omg*t)-1000*cos(90*

Figure 3-58. Using the Equation Editor to Build Amplitude Modulated Signal With Upper and Lower Sidebands

Combining Waveforms

The last but not least powerful feature allows you to combine waveforms, which you previously stored on your hard disc. You can write mathematical expressions that contain waveforms, simple operands and trigonometric functions similar to the example given below. If you want to use waveforms in your equations, you must first generate these waves and store them on your hard disk. You identify waveforms by adding the *.wav extension as shown in the example below.

Ampl(p)= Sine.wav*sin(omg*p*10)*Noise.wav/1000

The above equation will generate amplitude-modulated waveform with added noise. The following steps demonstrate how to create, store and combine waveforms using this equation.

Step 1 – Create and store sine.wav. Invoke the Equation Editor and type the following equation:

Ampl(p)= 5000*sin(omg*p)

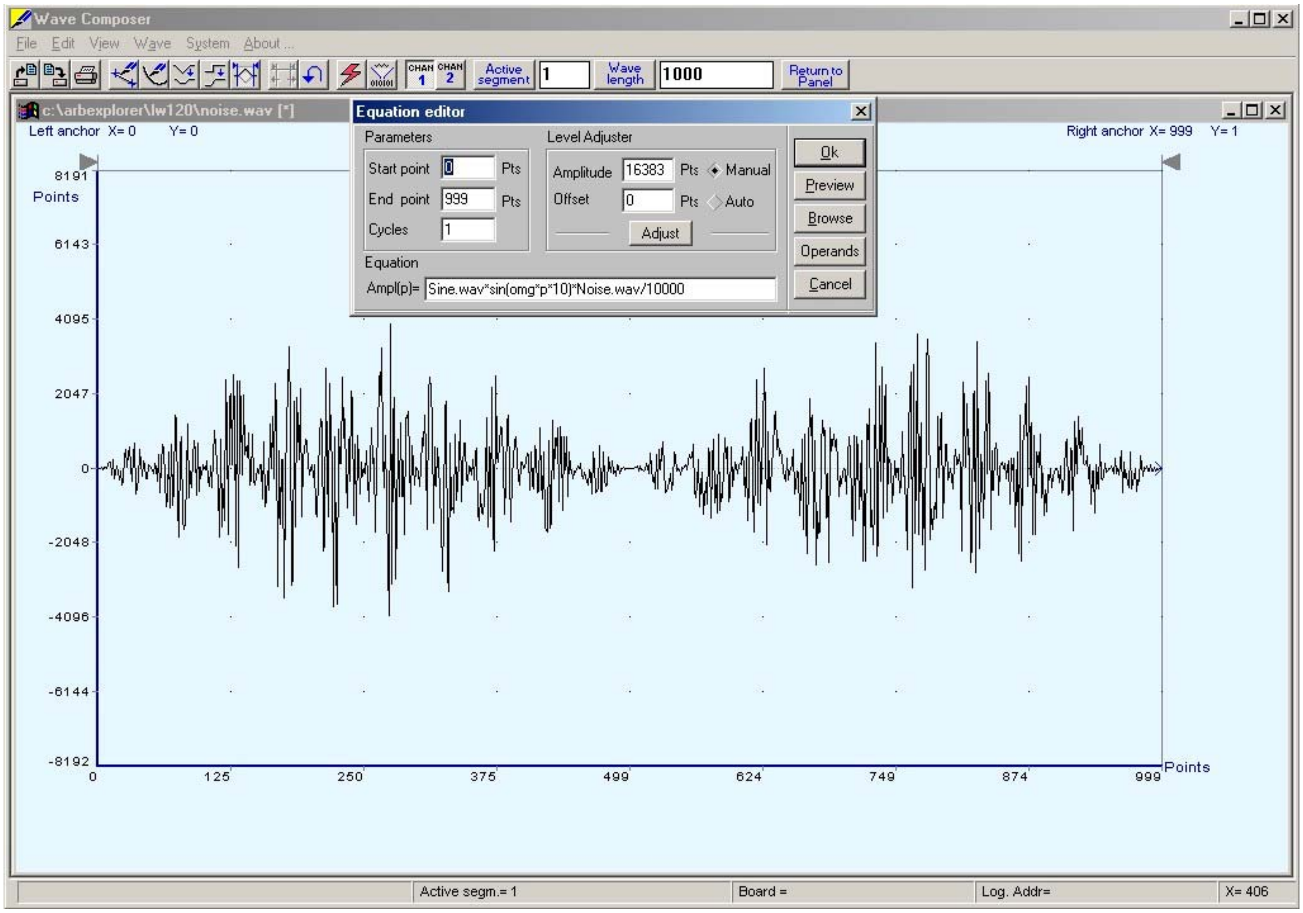
Press OK and then select the Save Waveform As… from the File command. Save this file using the name Sine.wav

Step 2 – Create and store Noise.wav. From the Wave command select Noise. Click OK and watch your waveform screen draw noisy signal. From the File menu select Save Waveform As… and save this waveform using the name Noise.wav.

Step 3 – Write and compute the original equation:

Ampl(p)= Sine.wav*sin(omg*p*10)*Noise.wav/1000

If you did not make any mistakes, your waveform screen should look as shown in Figure 3-29.



Equation editor

Parameters

Start point 0 Pts

End point 999 Pts

Cycles 1

Level Adjuster

Amplitude 16383 Pts ◆ Manual

Offset 0 Pts ◇ Auto

Equation

Ampl(p)= Sine.wav*sin(omg*p*10)*Noise.wav/10000

c:\arbexplorer\lw120\noise.wav [*]

Left anchor X= 0 Y= 0

Right anchor X= 999 Y= 1

Active segment 1

Wave length 1000

Active segm.= 1

Board =

Log. Addr=

X= 406

Figure 3-59. Combining Waveforms Into Equations

Creating FM Markers

Frequency markers are available at the SYNC output when the instrument is placed in either sweep or arbitrary FM modes. Unlike the normal SYNC operation where a SYNC bit is placed only once on a specific waveform segment, multiple frequency markers can be placed along the modulation curve thus, every time a specific frequency is hit, the SYNC outputs a marker pulse.

Marker position is programmed using the FM Markers Table, as shown in Figure 3-30. To program marker positions create your modulating waveform curve first and then, from the Edit menu select FM Markers…command. The FM Markers Table dialog box will show and will let you enter values in the edit field, on top of the markers list. Makers are entered in sample clock frequency units (SA/s) and should be within your deviation range. If you want to compute marker position in output frequency, then you should convert the units using the following equation:

Frequency = Sample clock frequency / n

where n = number of points of the modulating curve.

For example, if you look at Figure 3-30, the first marker position is 7.2006 MSa/s and the number of point on the horizontal axis is 1000 points (0 to 999 = 1000). The actual marker frequency is:

7.2006 MSa/s / 1000 = 7.2006 KHz

After you complete writing maker positions and if the specified markers are within the selected deviation range, the markers will be shown on the modulating curve as little blue dots. The various controls in the FM Markers Table are described below.

Sort – will automatically sort your markers in an ascending or descending order as you add them to the list.

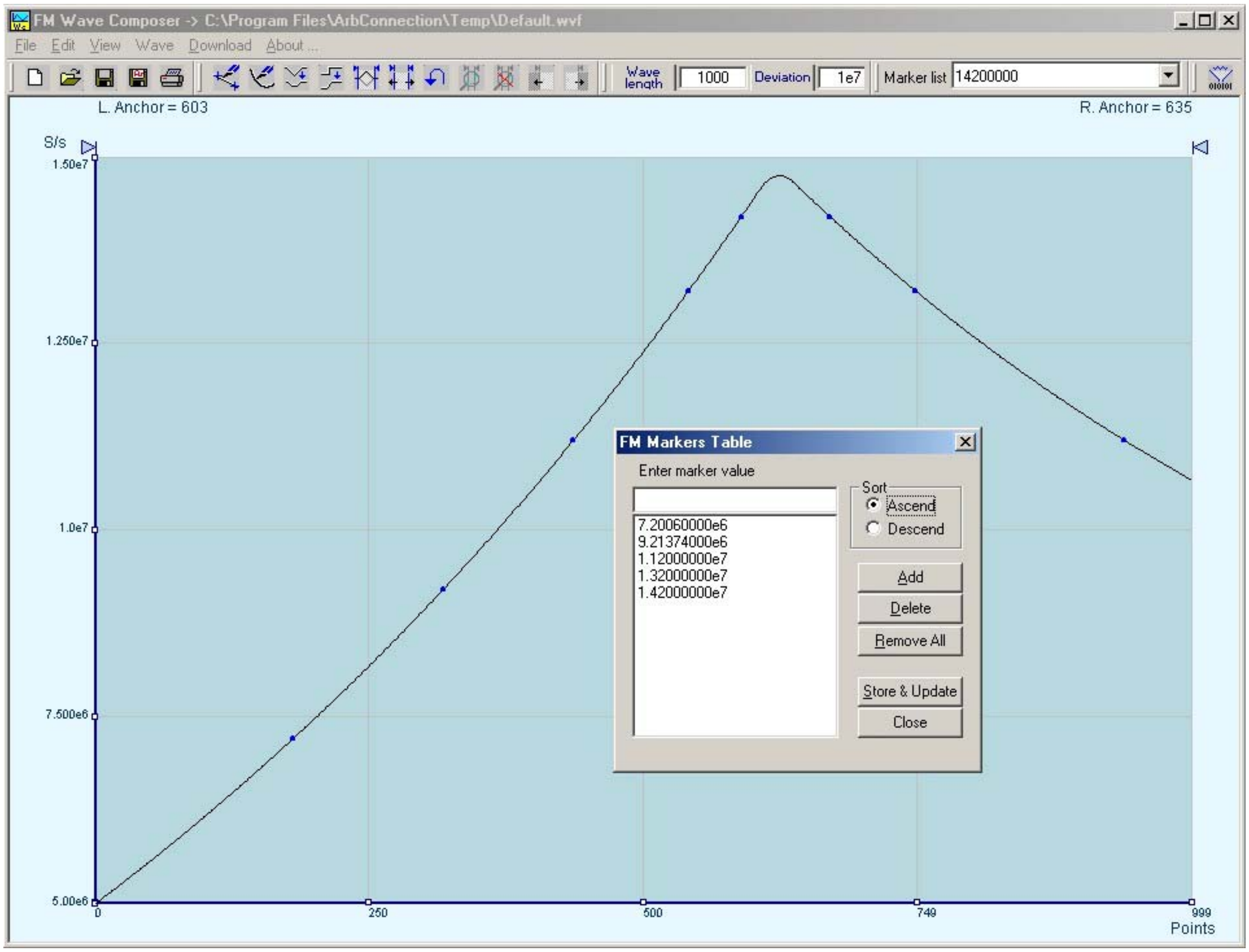
Add – will add a marker to the list after you type it in the Enter Marker Value field.

Delete – will delete a highlighted marker from the markers list.

Remove All – will remove all marker positions from the list and will clear the entire table.

Store & Update – will save your table and will update the program so you can close it safely without loosing table data.

Close – will remove the FM Markers Table from the screen and will update the modulating waveform with the marker positions that you assigned in the table. If you press Close without first Store & Update, you'll lose data entry. You can also use this button in case you want to discard of faulty data entries.



FM Wave Composer -> C:\Program Files\ArbConnection\Temp\Default.wvf

Wave length 1000 Deviation 1e7 Marker list 14200000

L. Anchor = 603

R. Anchor = 635

FM Markers Table

Enter marker value

7.20060000e6
9.21374000e6
1.12000000e7
1.32000000e7
1.42000000e7

Sort
● Ascend
○ Descend

Figure 3-60. Generating FM markers

This page intentionally left blank

Chapter 4

PROGRAMMING REFERENCE

What's In This Chapter

This Chapter lists and describes the software drivers that are supplied with the instrument. It also lists and describes the set of SCPI-compatible (Standard Commands for Programmable Instruments) commands used to operate the PXA125. To provide familiar formatting for users who have previously used the SCPI reference documentation, the command descriptions are dealt in a similar manner. In particular, each sub-system's documentation starts with a short description, followed by a table showing the complete set of commands in the sub-system; finally the effects of individual keywords and parameters are described. Complete listing of all commands used for programming the PXA125 is given in Table 4.1.

What's Required

In order for you to start programming the Model PXA125, you'll need to have some files, libraries and header files installed in various folders of your computer. Check that you have all of these files before you start with your own application.

When you run the install program from the supplied CD, the installation program will automatically generate a folder with the name PXA125. In this folder you'll have three other sub-folders: Bin, Include and Lib/MSC. Check in the Bin sub-folder that you have TEComm.dll and te5200drv.dll, check in the Include sub-folder that you have TEComm.h and te5200drv.h and check in the Lib/MSC folder that you have TEComm.lib and te5200drv.lib. Source code for an application example is also included in the main folder.

To program the PXA125, you'll need to copy the header and library files to your application folder. You'll also need the above two libraries and a set of SCPI commands. Description of the libraries and their functions is given below, followed by detailed introduction to SCPI format and syntax.

The TEComm.dll Functions

The TECom.dll controls communication with the PCI bus. There are five library functions that call this dll. These functions are described in the following.

FindInstrument

Purpose

Scans PXI baskets for PXA125 units and returns the number of instruments that it found.

C Syntax

INSTRUMENTSTATE FindInstrument (PDWORD pItems)

Resource Classes

PXI, PCI

Parameters

Name

Description

pItems

Returns the number of PXA125's in the system

Return Values

Completion Codes

Description

PCI_SUCCESS

Function executed without errors

Error Codes

Description

PCI_NFOUND_INS_IN_SYSTEM

No PXA125's were found in the system

Description

The FindInstrument() scans the PCI bus for Plug & Play instruments (PXA125/1's only) and detects how many instruments are plugged into your system. It also initiates communications and sets up file images with all of the found devices. Therefore, you must use this function in your program before you use any other functions.

The information that this function provides is handy for the OpenInstrument() function because it tells you how many instruments established valid communication link. Once you have this piece of information you can decide which one to open and use in your application.

The FindInstrument() function returns an integer. For example, if it finds one instrument only, then the pItems returns 0. Likewise, if there are 5 instruments in your system, then this pItems returns 4.

OpenInstruemnt

Purpose

Initiates communications with one of the instruments found using the function FindInstrument()

C Syntax

OpenInstrument (int hInstr)

Resource Classes

PXI, PCI

Parameters

Name

Description

hInstr

Instrument handle number. Must be out of a range of instruments that were found in the PXI baskets. The numbers start at 0 (for first device) to n-1. This value is extracted using the TEComm.dll function FindInstrument()

Return Values

Completion Codes

Description

PCI_SUCCESS

Function executed without errors

PCI_INS_ALREADY_OPEN

Trying to establish new communication link while instrument has already been connected.

Error Codes

Description

PCI_ERR_OPEN_INS_FAILED

Could not succeed to initiate communications with the selected instrument.

Description

The OpenInstrument() function can be used only after you initiate communication with your instruments using the function FindInstrument(). This function enables communication and assigns a handle to specific device number. For example, if you want to use device number 2, set the value of hInstr to 1. Likewise, if you want to use device number 8, set the value of hInstr to 7.

OpenAllInstrument

Purpose

Initiates communications with all instruments found using the function FindInstrument()

C Syntax

OpenAllInstrument (void)

Resource Classes

PXI, PCI

Parameters

None

Return Values

Completion Codes

Description

PCI_SUCCESS

Function executed without errors

PCI_INS_ALREADY_OPEN

Trying to establish new communication link while instrument has already been connected.

Error Codes

Description

PCI_ERR_OPEN_INS_FAILED

Could not succeed to initiate communications with the selected instrument.

Description

The OpenAllInstrument() function can be used only after you initiate communication with your instruments using the function FindInstrument(). This function enables communication and assigns handles to all of the instruments at once. In this case, you do not have to specify device numbers.

CloseInstrument

Purpose

Terminates communications with one of the instruments

C Syntax

CloseInstrument (int hInstr)

Resource Classes

PXI, PCI

Parameters

Name

Description

hInstr

Instrument handle number. Must be out of a range of instruments that were found in the PXI baskets. The numbers start at 0 (for first device) to n-1. This value is extracted using the TEComm.dll function FindInstrument()

Return Values

Completion Codes

Description

PCI_SUCCESS

Function executed without errors

PCI_INS_ALREADY_OPEN

Trying to establish new communication link while instrument has already been connected.

Error Codes

Description

PCI_ERR_INDEX_OF_INS_NOT_EXIST

Illegal index number used to identify an instrument

Description

Use this function if you want to remove the handle from one or more instruments. After you free the handle, you will not be able to access the instrument before you use the Openinstrument() function again.

GetInstrumentAttribute

Purpose

Use this function to fetch instrument identity, options, slot location, serial number etc.

C Syntax

```
GetInstrumentAttribute (int hInstr, INSTRATTR* pInstrAttr)
```

Resource Classes

PXI, PCI

Parameters

Name

Description

hInstr

Instrument handle number. Must be out of a range of instruments that were found in the PXI baskets. The numbers start at 0 (for first device) to n-1. This value is extracted using the TEComm.dll function FindInstrument()

pInstrAttr

Attributes structure. The structure is available in TEComm.h (INSTRATTR).

Return Values

Completion Codes

Description

PCI_SUCCESS

Function executed without errors

PCI_INS_ALREADY_OPEN

Trying to establish new communication link while instrument has already been connected.

Error Codes

Description

PCI_ERR_INDEX_OF_INS_NOT_EXIST

Illegal index number used to identify an instrument

Description

The purpose of this function is to allow you to read instrument identity and instrument options. There are many parameters that this function can supply. These parameters are listed in the TEComm.h header file. Description of the structure is given in the following.

GetInstrumentAttribute (continued)

Instrument Attributes Structure - INSTRATTR

m_Type Describes the model number. For example – PXA125

m_Description Provides description of the instrument – Arbitrary Waveform Generator

m_BasicModel Provides in formation on maximum sample clock frequency. 125 MS/s for the PXA125.

m_MemSize Provides information on the installed memory. There are 2 Megs available.

m_ClockOption Provides information on TCXO. The return will be TCXO.

m_Customer Provides information on the name of the brand this instrument was labeled for. For example, Team Marketing

m_Date Provides information on the date the instrument was first turned on and its flash memory loaded with the driver.

m_Rev Provides information on the hardware revision number of the instrument. Revision number is given in letters, i.e., A, B, etc.

m_SerialNumber Provides the serial number of the instrument

m_Platf Provides information on the Interface platform. For example, PXI, PCI, cPCI etc.

m_VirtualSlot Provides information on slot location of the instrument. The virtual slot location must be defined by the user, using the Arb Detector from the Startup menu.

m_NumChassis In case you are using multiple chassis system, this parameter lets you identify the chassis number of which the instrument is plugged into. The chassis number and the slot location must be defined by the user, using the Arb Detector from the Startup menu.

Instrument Attributes Structure - INSTRPHISICALLOCATION

m_Bus Provides information on the bus number the instrument was found on.

m_Slot Provides information on the slot location that was assigned by the computer. Note that physical location and logical location are not the same. You can program the virtual slot location for your own convenience but the physical slot is assigned by the computer and can not be modified for your application

Instrument Attributes Structure - INSTRID

m_Vendor Provides vendor ID number

m_DeviceID Provides device ID number

GetStateMessage

Purpose

Returns a string with a description of the error found

C Syntax

Void GetStateMessage (INSTRUMENT state, char *pMsg, int size)

Resource Classes

PXI, PCI

Parameters

Name

Description

state

An error code. The error list is available in TEComm.h.

pMsg

Returns a string identifying the error

Size

Assigned buffer size for the returned string

Return Values

None

Description

Use this function to detect errors while sending commands to the instrument. Errors could generate from not being able to communicate with the devices, or from sending illegal SCPI syntax. Setting conflicts are also common source for errors. The GetStateMessage() function returns codes for errors. These codes are interpreted in the TEComm.h header file.

The TE5200drv.dll Functions

The TE5200drv.dll controls the complete functionality of the instrument. You also have to use this dll to communicate with your instrument. Communication with the instrument is done using SCPI commands. The SCPI commands by themselves cannot program the instrument but must be used with conjunction with the TE5200drv.dll. There are two library functions that call this dll. These functions are described in the following..

SendCommand

Purpose

Use this function to send SCPI commands to the instrument

C Syntax

INSTRUMENTSTATE SendCommand (int hInstr, char *inbuf)

Resource Classes

PXI, PCI

Parameters

Name

Description

hInstr

Instrument handle number. Must be out of a range of instruments that were found in the PXI baskets. The numbers start at 0 (for first device) to n-1. This value is extracted using the TEComm.dll function FindInstrument()

inbuf

Contains one or more SCPI commands. 256 characters limit the number of characters in this buffer. The commands must be terminated by '\0'. If the inbuf contains queries, the returned strings will be placed in the inbuf and also in this case, the size is limited by 256 characters.

Return Values

Completion Codes

Description

PCI_SUCCESS

Function executed without errors

PCI_INS_ALREADY_OPEN

Trying to establish new communication link while instrument has already been connected.

Error Codes

Description

PXI_COMMAND_ERROR

if value is 0x8000000, error is in SCPI syntax or other errors relating to instrument setability .SCPI error list is given in the SCPI Commands reference.

If value is larger than 0x8000000, look for problems in the communication with the PXA125. Most probable cause for such errors is not invoking the Arb Detector program after you power up your computer.

Description

The SendCommand() function is the main function you'll use for controlling instrument functions and parameters. The PXA125 is using Standard Commands for Programmable Instruments (SCPI). The following paragraphs describe SCPI syntax, structure and format. In general, you can program the instrument by sending one command at a time. However, the inbuf has a capacity to accept up to 256-character strings.

Using SCPI language should look familiar to programmers that use to program GPIB or VXI instruments. The command structure and syntax is exactly the same, which will make it even easier to program the PXA125. However, do not get discouraged if you are not familiar with SCPI programming. SCPI introduction and command examples are given later in this chapter. If you feel that the introduction in this manual is not enough for you, you may consider purchasing the SCPI specification book from the SCPI consortium.

A simple example of the SendCommand() is given below:

```
SendCommand(0, outp 1;:outp:sync on;:func:shap squ;:volt 1;offs 0.5;:freq 1.234e6)
```

This single command line is equivalent to a series of commands as shown below:

```
SendCommand(0, outp 1)              \\ Turns the main output ON

SendCommand(0, outp:sync on)                 \\ Turns the SYNC output ON

SendCommand(0, func:shap squ)       \\ Modifies output waveform shape to square

SendCommand(0, volt1;offs0.5)       \\ Modifies amplitude to 1V and offset to 0.5V

SendCommand(0, freq 1.234e6)        \\ Modifies frequency to 1.234 MHz
```

Hints

1. To get you started quickly with SCPI programming and without the hassle of learning the syntax, you'd better start with series of commands and later, as you get familiar with the programming aspects, you can start using shorted command lines.

2. Later in this chapter, you'll find all of the commands that are used for programming the instrument. This reference is the complete and only source you'll need for your application. However, if you want to save some time and not study the complete chapter, you can use the log feature in ArbConnection to view commands as you click on buttons and change parameters and then simply copy these commands into your application. To access the log file, invoke ArbConnection select the View menu and click on Log File. Inside the Log File View check the Show Commands Only. From this point, every time you change anything on the panel, the action will be recorded in the log file using the exact syntax as you should use in your application.

3. You can use the SendCommand() function to download arbitrary waveform data to the memory. In this case, you'll have to prepare an external file that has the waveform data in binary format and concatenate the file name with the trace# command. The file structure for the waveform data is shown in the SendBlock() command description.

SendBlock

Purpose

Use this function to send arbitrary waveform coordinates as well as FM arbitrary waveform coordinates to the instrument

C Syntax

INSTRUMENTSTATE SendBlock (int hInstr, WORD mode, char *pFileName, void *pWave, DWORD size)

Resource Classes

PXI, PCI

Parameters

Name

Description

hInstr

Instrument handle number. Must be out of a range of instruments that were found in the PXI baskets. The numbers start at 0 (for first device) to n-1. This value is extracted using the TEComm.dll function FindInstrument()

pFileName

Use this parameter to define if the waveform coordinates are located in an external file or reside inside the program. "NULL" define resident coordinates. "file_name>" define name and path of external file. You can use complete path if file location is outside the working folder.

pWave

Pointer for memory space where the coordinates of the arbitrary waveform can be found. If you are using external files, then use NULL for this parameter.

size

If you are using an external file for waveform coordinates, then leave this parameter 0. If you are using resident coordinates, this parameter specifies the number of bytes at the pointer location that are used as coordinates for the arbitrary waveform. For arbitrary waveforms, each data point requires

2 bytes and for arbitrary FM waveforms, each data point requires 5 bytes.

Return Values

Completion Codes

Description

PCI_SUCCESS

Function executed without errors

PCI_INS_ALREADY_OPEN

Trying to establish new communication link while instrument has already been connected.

Error Codes

Description

PXI_COMMAND_ERROR

if value is 0x8000000, error is in SCPI syntax or other errors relating to instrument setability .SCPI error list is given in the SCPI Commands reference.

If value is larger than 0x8000000, look for problems in the communication with the PXA125. Most probable cause for such errors is not invoking the Arb Detector program after you power up your computer.

Description

The PXA125 is a waveform generator. There are three types of waveforms that the instrument can generate: 1) Standard waveforms, 2) Arbitrary waveforms and 3) Waveforms that are frequency modulated by arbitrary modulating waveforms. Every time that the instrument is being asked to generate waveforms, it must have waveform coordinates in its working memory.

The generator has nine built-in standard waveforms shapes. There is no need to download waveform coordinates to the generator for it to be able to output one of these waveforms. However, every time you select one of the standard functions, the firmware computes the data and places the waveform coordinates in the working memory.

Not so if you want to generates arbitrary waveforms. In this case, you have to generate the coordinates from an external application and then download the data to the generator. There are two separate memory banks built into the instrument. The main and the largest memory bank generates the arbitrary (USER) waveforms. This memory has vertical resolution of 14 bits and depth (or length) of 2 Meg. The other memory bank is used for arbitrary waveforms that modulate the carrier frequency. This bank is completely different than the main memory bank in the way that it contains data. Data structure for both memory types is given hereinafter.

The SendBlock() function is used for sending arbitrary blocks of data to the instrument. Here are some simple steps for you to follow in your application. Two examples are given: Using the SendBlock() function with resident data array and using SendCommand() with an external data file.

Example 1

1. Use SendCommand to program Trac:sel 1
2. Use SendCommand to program Trac:def 1,1000
3. Prepare your waveform data
4. Call SendBlock(hInstr, 1, "", (void*)wav, 1000*2). Note that each data point requires 2 bytes.

Example 2

1. Use SendCommand to program Trac:sel 1
2. Use SendCommand to program Trac:def 1,1000
3. Prepare your waveform data and save in a file. Concatenate the file name with trace#
4. Call SendCommand(0,trac#<file_name>)

Arbitrary Waveform Data Structure

The arbitrary waveform data structure is actually very simple. It contains 16-bit words in the range of –8192 to 8291, converted to short format (Intel binary representation)

Arbitrary FM Waveform Data Structure

The arbitrary FM waveform data structure is a bit more complicated. It is made of sample clock frequencies and not as coordinates as was the structure for the arbitrary waveform data. Sample clock range is from 50 S/s to over 100 MS/s. The arbitrary FM waveform memory has 200K points maximum however, you can choose to use shorter waveform length by specifying the "size" parameter in the SendBlock() function.

Here are some simple steps that show how to prepare the arbitrary FM waveform data.

1. Define a structure:

```
Struct Fmtag( int pt;
     char last;
}FMPT;
```

2. Define array FMPT fmWav[<wavelength>]
3. Fill the structure with data array of which each fmWav represents:

   pt – S/s value x 14.31655765 (S/s is limited by specified sample clock range)

   last – Ored value 0x80 for the last waveform data point
   0x01 for marker position
   0x00 for all other points
4. Convert pt component of array to long format (Intel binary representation, if necessary)
5. Call DLL function SendBlock(hInstr, 2, "", (void*)fmWav, <wavelength>*5)

Note that the arbitrary FM waveform data array is comprised of 5 bytes. The most significant 4 bytes contain waveform data. The lease significant byte ("last") flags marker position and end of data file. Therefore, each arbitrary FM waveform data point is represented by 5 bytes.

Introduction To SCPI

Commands to program the instrument over the GPIB are defined by the SCPI 1993.0 standard. The SCPI standard defines a common language protocol. It goes one step further than IEEE-STD-488.2 and defines a standard set of commands to control every programmable aspect of the instrument. It also defines the format of command parameters and the format of values returned by the instrument.

SCPI is an ASCII-based instrument command language designed for test and measurement instruments. SCPI commands are based on a hierarchical structure known as a tree system. In this system, associated commands are grouped together under a common node or root, thus forming subsystems.

Part of the OUTPut subsystem is shown below to illustrate the tree system:

```
:OUTPut
  :FILTer
     [:LPASs]                {NONE|25M|50M|ALL}
        [:STATe] OFF | ON
```

OUTPut is the root keyword of the command; FILTer and STATe are second level keywords. LPASs is third level keyword. A colon (:) separates a command keyword from a lower level keyword.

Command Format

The format used to show commands in this manual is shown below:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

The command syntax shows most commands (and some parameters) as a mixture of upper and lowercase letters. The uppercase letters indicate the abbreviated spelling for the command. For shorter program lines, send the abbreviated form. For better program readability, use the long form.

For example, in the above syntax statement, FREQ and FREQUENCY are both acceptable forms. Use upper or lowercase letters. Therefore, FREQ, FREQUENCY, freq, and Freq are all acceptable. Other forms such as FRE and FREQUEN will generate an error.

The above syntax statement shows the frequency parameter enclosed in triangular brackets. The brackets are not sent with the command string. A value for the frequency parameter (such as "FREQ 50e+6") must be specified.

Some parameters are enclosed in square brackets ([]). The brackets indicate that the parameter is optional and can be omitted. The brackets are not sent with the command string.

Command Separator

A colon (:) is used to separate a command keyword from a lower level keyword as shown below:

```
SOUR:FUNC:SHAP SIN
```

A semicolon (;) is used to separate commands within the same subsystem, and can also minimize typing. For example, sending the following command string:

```
TRIG:SLOP NEG;COUN 10;TIM 5e-3
```

is the same as sending the following three commands:

```
:TRIG:SLOP                                NEG
:TRIG:COUN                                 10
:TRIG:TIM 5e-3
```

Use the colon and semicolon to link commands from different subsystems. For example, in the following command string, an error is generated if both the colon and the semicolon are not used.

```
OUTP:STATE ON;:TRIG:BURS ON
```

The MIN and MAX Parameters

Substitute MINimum or MAXimum in place of a parameter for some commands. For example, consider the following command:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

Instead of selecting a specific frequency, substitute MIN to set the frequency to its minimum value or MAX to set the frequency to its maximum value.

Querying Parameter Setting

Query the current value of most parameters by adding a question mark (?) to the command. For example, the following command sets the output function to square:

```
SOUR:FUNC:SHAP SQR
```

Query the output function by executing:

```
SOUR:FUNC:SHAP?
```

Query Response Format

The response to a query depends on the format of the command. In general, a response to a query contains current values or settings of the generator. Commands that set values can be queried for their current value. Commands that set modes of operation can be queried for their current mode. IEEE-STD-488.2 common queries generate responses, which are common to all IEEE-STD-488.2 compatible instruments.

SCPI Command Terminator

A command string sent to the function generator must terminate with a <new line> character. The IEEE-STD-488 EOI message is a <new line> character. Command string termination always resets the current SCPI command path to the root level.

IEEE-STD-488.2 Common Commands

The IEEE-STD-488.2 standard defines a set of common commands that perform functions like reset, trigger and status operations. Common commands begin with an asterisk (*), are four to five characters in length, and may include one or more parameters. The command keyword is separated from the first parameter by a blank space. Use a semicolon (;) to separate multiple commands as shown below:

```
*RST; *STB?; *IDN?
```

SCPI Parameter Type

The SCPI language defines four different data formats to be used in program messages and response messages: numeric, discrete, boolean, and arbitrary block.

Numeric Parameters

Commands that require numeric parameters will accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. Special values for numeric parameters like MINimum and MAXimum are also accepted.

Engineering unit suffices with numeric parameters (e.g., MHz or kHz) can also be sent. If only specific numeric values are accepted, the function generator will ignore values, which are not allowed and will generate an error message. The following command is an example of a command that uses a numeric parameter:

```
VOLT:AMPL <amplitude>
```

Discrete Parameters

Discrete parameters are used to program settings that have a limited number of values (i.e., FIXed, USER and SEQuence). They have short and long form command keywords. Upper and lowercase letters can be mixed. Query responses always return the short form in all uppercase letters. The following command uses discrete parameters:

```
SOUR:FUNC:MODE {FIXed | USER | SEQuence}
```

Boolean Parameters

Boolean parameters represent a single binary condition that is either true or false. The generator accepts "OFF" or "0" for a false condition. The generator accepts "ON" or "1" for a true condition. The instrument always returns "0" or "1" when a boolean setting is queried. The following command uses a boolean parameter:

```
OUTP:FILT { OFF | ON }
```

The same command can also be written as follows:

```
OUTP:FILT {0 | 1 }
```

SCPI Syntax and Styles

Where possible the syntax and styles used in this section follow those defined by the SCPI consortium. The commands on the following pages are broken into three columns; the KEYWORD, the PARAMETER FORM, and any NOTES.

The KEYWORD column provides the name of the command. The actual command consists of one or more keywords since SCPI commands are based on a hierarchical structure, also known as the tree system. Square brackets ([ ]) are used to enclose a **keyword** that is optional when programming the command; that is, the PXA125 will process the command to have the same effect whether the optional node is omitted by the programmer or not. Letter case in tables is used to differentiate between the accepted short form (upper case) and the long form (upper and lower case).

The PARAMETER FORM column indicates the number and order of parameter in a command and their legal value. Parameter types are distinguished by enclosing the type in angle brackets (< >). If **parameter** form is enclosed by square brackets ([ ]) these are then optional (care must be taken to ensure that optional parameters are consistent with the intention of the associated keywords). The vertical bar (|) can be read as "or" and is used to separate alternative parameter options.

Table 4-1. Model PXA125 SCPI Commands List Summary

Keyword	Parameter Form (Default in Bold)	Notes
:INSTRument		
:COUPle		
[:STATe]	**OFF** | ON	
:MODE	**MASTer** | SLAVe	
:PHASe		
[:OFFSet]	(**0**,0,n) where “n” is the size of the active segment	4 point increments
:SET	0 | 1 | 2	
:OUTPut		
[:STATe]	**OFF** | ON	
:FILTer		
[:LPASs]	**NONE** | 25M | 50M | ALL	
:TTLTrig<N>		N ranges from 0 to 7
[:STATe]	**OFF** | ON	
:STAR		
[:STATe]	**OFF** | ON	
:SYNC		
[:STATe]	**OFF** | ON	
:SOURce	**BIT** | LCOMplete	
:POSition		
[:POINt]	(**0**;0;n) where “n” is the size of the active segment	4 point increments
:WIDTh	(4;4;100000)	4 point increments
[:SOURce]		
:APPLy		
:SINusoid	FREQ,AMPL,OFFS,PHAS	
:TRIangle	FREQ,AMPL,OFFS,PHAS	
:SQUare	FREQ,AMPL,OFFS,DCYC	
:PULSe	FREQ,AMPL,OFFS,DEL,WIDT,LEAD,TRA	
:RAMP	FREQ,AMPL,OFFS,DEL,LEAD,TRA	
:SINC	FREQ,AMPL,OFFS, NCYC	
: GAUSsian	FREQ,AMPL,OFFS,EXP	
: EXPonential	FREQ,AMPL,OFFS,EXP	
: DC	DC_AMPL	
: USER	SEGM,SCLK,AMPL,OFFS	
:FREQuency		
[:CW]	(**1e6**;100e-6;50e6) | MINimum | MAXimum	
:RASTer	(**1e7**;100e-3;100e6) | MINimum | MAXimum	
:SOURce	INTernal | EXTernal	

Table 4-1. Model PXA125 SCPI Commands List Summary (continued)

Keyword	Parameter Form (Default in Bold)	Notes
:ROSCillator		
:SOURce	**INTernal** | EXTernal | TCXO	
:VOLTage		
[:LEVel]		
[:AMPLitude]	(5**.000**;80E-3;8.00) | MINimum | MAXimum	
:OFFSet	(**0**;-3.6;+3.6)	
:FUNCtion		
:MODE	**FIXed** | USER | SEQuence	
:SHAPe	**SINusoid** | TRIangle | SQUare | PULSe | RAMP |	
	SINC | GAUSsian | EXPonential | NOISe | DC	
:SINusoid		
:PHASe	(**0**;0;360)	
:TRIangle		
:PHASe	(**0**;0;360)	
:SQUare		
:DCYCle	(**50**;1;99)	
:PULSe		
:DELay	(**10**;0;99.9)	
:WIDth	(**10**;0;99.9)	
:TRANsition		
[:LEADing]	(**10**;0;99.9)	
:TRAiling	(**10**;0;99.9)	
:RAMP		
:DELay	(**0**;0;99.9)	
:TRANsition		
[:LEADing]	(6**0**;0;99.9)	
:TRAiling	(**30**;0;99.9)	
:GAUSsian		
:EXPonent	(2**0**;10;200)	
:SINC		
:NCYCle	(**10**;4;100)	
:EXPonential		
:EXPonent	(**1**;-20.00;20.00)	
:DC		
[:AMPLitude]	(**100**;-100;100)	

Table 4-1. Model PXA125 SCPI Commands List Summary (continued)

Keyword	Parameter Form (Default in Bold)	Notes
:FM		
:DEViation	(**10e6**;100e-3;100e6)	
:FUNCtion		
:MODE	**FIXed** | USER	
:SHAPe	**SINusoid** | TRIangle | SQUare | RAMP	
:FREQuency	(**1e3**;1e-3;100e3)	
:RASTer	(**1e6**;1e-3;10e6)	
[:STATe]	**OFF** | ON	
:TRIGger		
:MODE	**CONTinuous** | TRIGgered | GATed	
:SLOPe	**POSitive** | NEGative	
:FSK		
:FREQuency		
:RASTer	(**80e6**;100e-3;100e6)	
[:STATe]	**OFF** | ON	
:MODE	**HOP** | RAMP	
:RAMP		
:TIME	(1e-3;10e-6;1)	
:SWEep		
[:FREQuency]		
:STOP	(**20e6**;100e-3;100e6)	
[:STATe]	**OFF** | ON	
:TIME	(**1e-3**;1e-3;1000)	
:DIRection	**UP** | DOWN	
:SPACing	**LINear** | LOGarithmic	
:TRIGger	**CONTinuous** | TRIGgered | GATed	
:MODE	**CONTinuous** | TRIGgered | GATed	
:SLOPe	**POSitive** | NEGative	
:MARKer	(20e6;100e-3;100e6)	
:TRACe		
DEFine	(1;1;4096),(16;16;2097152)	Even number, divisible by 4
:DELete		
[:NAME]	(1;1;4096)	
:ALL		
:SELect	(**1**;1;4096)	

Table 4-1. Model PXA125 SCPI Commands List Summary (continued)

Keyword	Parameter Form (Default in Bold)	Notes
:SEQuence		
[:DATA]	<binary_block>	
:ADVance	**AUTOmatic** | STEP | SINGle | MIXed	
:SOURce	**EXTernal** | INTernal |TTLTrig 0-7 | STAR	
:DEFine	(1;1;4096),(1;1;4096),(1;1;128K),(0,0,1)	Step, segment, repeat, advance mode
:DELete		
:ALL		
:INITiate		
[:IMMediately]		
:CONTinuous	**ON** | OFF	
:TRIGger		
:BURSt		
[:STATe]	**OFF** | ON	
:COUNt	(**1**;1;1E6)	
:SOURce		
:ADVance	**EXTernal** | INTernal | TTLTrig 0-7 | STAR	
:GATE		
[:STATe]	**OFF** | ON	
:SLOPe	**POSitive** | NEGative	
:TIMer	(**1e3**;100e-3;5e6)	In Hz units
:STAR		
[:STATe]	**OFF** | ON	
:PHASe	(**0**;0;n) where “n” is the size of the active segment	4 points increments
:ARM		
[:STATe]	**OFF** | ON	
:SLOPe		
[:STARt]	**POSitive** | NEGative	
:BREakpoint		
:POSition	(**0**;0;999999)	4 points increments
:RESet		
:SYSTem		
:ERRor?		Query only
:VERSion?		Query only, 1999.0
:IDENtification		Query only

Table 4-1. Model PXA125 SCPI Commands List Summary (continued)

Keyword	Parameter Form (Default in Bold)	Notes
*CLS		
*ESE	(**0**;0;255)	
*OPC		
*RST		
*SRE	(**0**;0;255)	
*TRG		
*ESE?		Query only
*ESR?		Query only
*IDN?		Query only
*OPC?		Query only
*OPT?		Query only
*SRE?		Query only
*STB?		Query only

SOURce Subsystem

This subsystem is used to control output functions, shape and parameters, frequency, amplitude and amplitude modulation, and filter type. Optional nodes were omitted from these commands. Factory defaults after *RST are shown in bold typeface. Parameter low and high limits are given where applicable.

Keyword **Parameter Form** **Default,**Low Limit,High Limit

```
[:SOURce]
  :APPLy (?)
    :SINusoid (?)      {<freq>,<ampl>,<offs>,<phase>}
    :TRIangle (?)      {<freq>,<ampl>,<offs>,<phase>}
    :SQUare (?)        {<freq>,<ampl>,<offs>,<duty_cycle>}
    :PULSe (?)         {<freq>,<ampl>,<offs>,<delay>,<rise>,<high>,<fall>}
    :RAMP (?)          {<freq>,<ampl>,<offs>,<delay>,<rise>,<fall>}
    :SINC (?)          {<freq>,<ampl>,<offs>,<N_cycles>}
    :EXPonential (?)   {<freq>,<ampl>,<offs>,<exp>}
    :GAUSsian (?)      {<freq>,<ampl>,<offs>,<exp>}
    :DC (?)            <%_ampl>
    :USER (?)          {<seg#>,<sclk>,<ampl>,<offs>}
  :FM  (?)             {OFF|ON|0|1}                          OFF
    :DEViation (?)     <deviation>                           1e6,100e-3,100e6
    :FUNCtion
      :MODE (?)        {FIXed|USER}                          FIXed
      :SHAPe (?)       {SINusoid|TRIangle|SQUare|RAMP)       SINusoid
    :FREQuency         <FM_freq>                             1e3,1e-3,100e3
      :RASTer (?)      <FM_sclk>                             1e6,1e-3,10e6
    :TRIGger
      :MODE (?)        {CONTinuous|TRIGgered|GATed}          CONTinuous
      :SLOPe (?)       {POSitive|NEGative                    POSitive
  :FREQuency (?)       {<freq>|MINimum|MAXimum}              1e6,100e-6,50e6
    :RASTer (?)        {<sclk>|MINimum|MAXimum}              10e6,100e-3,100e6
      :SOURce (?)      {EXTernal|INTernal}                   INTernal
  :FSK (?)             OFF|ON|0|1}                           OFF
    :FREQuency
      :RASTer (?)      <FSK_sclk>                            80e6,100e-3,100e6
    :MODE (?)          {HOP|RAMP}                            HOP
    :RAMP
      :TIME (?)        <time>                                1e-3,10e-6,1
  :FUNCTion
    :MODE (?)          {FIXed|USER|SEQuence}                 FIXed
    :SHAPe (?)         {SINusoid|TRIangle|SQUare|PULSe|RAMP
                       |SINC|EXPonential|GAUSsian|NOISe|DC}SINusoid
  :ROCSillator
    :SOURce (?)        (INTernal|EXTernal | TCXO)            INTernal
  :SWEep (?)           OFF|ON|0|1}                           OFF
    :STOP (?)          <stop_sclk>                           20e6,100e-3,100e6
    :TIME (?)          <time>                                1e-3,1e-3,1000
    :DIRection (?)     (UP|DONW)                             UP
    :SPACing (?)       (LINear|LOGarithmic)                  LINear
    :TRIGger
    :MODE (?)          {CONTinuous|TRIGgered|GATed}          CONTinuous
    :SLOPe (?)         {POSitive|NEGative                    POSitive
```

```
   :MARKer (?)          <mark_sclk>                           64e6,100e-3,100e6
:VOLTage (?)            {<ampl>|MINimum|MAXimum}              5,10e-3,10
     :OFFSet (?)        <offs>                                0,-3.6,3.6
   :SINusoid
     :PHASe (?)         <phase>                               0,0,360
   :TRIangle
     :PHASe (?)         <phase>                               0,0,360
   :SQUare
     :DCYCle (?)        <duty_cycle>                          50,1,99
   :PULSe
     :DELay (?)         <delay>                               10,0,99.9
     :WIDTh (?)         <pulse_width>                         10,0,99.9
     :TRANsition (?)    <rise>                                10,0,99.9
       :TRAiling (?)    <fall>                                10,0,99.9
   :RAMP
     :DELay (?)         <delay>                               0,0,99.9
     :TRANsition (?)    <rise>                                60,0,99.9
       :TRAiling (?)    <fall>                                30,0,99.9
   :SINC
     :NCYCle (?)        N_cycles>                             10,4,100
   :GAUSsian
     :EXPonent (?)      <exp>                                 20,10,200
   :EXPonential
     :EXPonent (?)      <exp>                                 -20,-100,+100
   :DC (?)              <%_amplitude>                         100,-100,+100
```

APPLy:SINusoid{<freq>,<ampl>,<offs>,<phase>}

Purpose

This command is a high-level command that programs the PXA125 to output sine waveform along with its associated parameters

- <freq> sets the output frequency in units of hertz
- <ampl> sets the output amplitude in units of volts
- <offs> sets the output offset in units of volts
- <phase> sets the output start phase in units of degrees

Parameter type

Numeric

Parameter range

<freq>	10e3 to 50e6
<ampl>	80e-3 to 8
<offs>	-3.6 to +3.6
<phase>	0 to 360

APPLy:SINusoid?

Response

The PXA125 will return <freq>, <ampl>,<offs>,<phase>. Default values are 1e6,5,0,0

APPLy:TRIangle{<freq>,<ampl>,<offs>,<phase>}

Purpose
This command is a high-level command that programs the PXA125 to output triangular waveform along with its associated parameters

- <freq> sets the output frequency in units of hertz
- <ampl> sets the output amplitude in units of volts
- <offs> sets the output offset in units of volts
- <phase> sets the output start phase in units of degrees

Parameter type
Numeric

Parameter range

<freq>	10e3 to 50e6
<ampl>	80e-3 to 80
<offs>	-3.6 to +3.6
<phase>	0 to 360

APPLy:TRIangle?

Response
The PXA125 will return <freq>, <ampl>,<offs>,<phase>. Default values are 1e6,5,0,0

APPLy:SQUare{<freq>,<ampl>,<offs>,<duty_cycle>}

Purpose
This command is a high-level command that programs the PXA125 to output square waveform along with its associated parameters

- <freq> sets the output frequency in units of hertz
- <ampl> sets the output amplitude in units of volts
- <offs> sets the output offset in units of volts
- <duty_cycle> sets the squarewave duty cycle in units of %

Parameter type
Numeric

Parameter range

<freq>	10e3 to 50e6
<ampl>	80e-3 to 8
<offs>	-3.6 to +3.6
<duty_cycle>	0 to 360

APPLy:SQUare?

Response
The PXA125 will return <freq>,<ampl>,<offs>,<duty_cycle>. Default values are 1e6,5,0,50

APPLy:PULSe{<freq>,<ampl>,<offs>,<delay>,<rise>,<high>,<fall>}

Purpose
This command is a high-level command that programs the PXA125 to output pulse waveform along with its associated parameters

- <freq> sets the output frequency in units of hertz
- <ampl> sets the output amplitude in units of volts
- <offs> sets the output offset in units of volts
- <delay> sets the delay to the pulse rise transition in units of %
- <rise> sets the pulse rise time in units of %
- <high> sets the pulse high time in units of %
- <fall> sets the pulse fall time in units of %

Parameter type

Numeric

Parameter range

<freq>	10e3 to 50e6
<ampl>	80e-3 to 8
<offs>	-3.6 to +3.6
<delay>	0 to 99.9
<rise>	0 to 99.9
<high>	0 to 99.9
<fall>	0 to 99.9

APPLy:PULSe?

Response

The PXA125 will return <freq>, <ampl>,<offs>,<delay>,<rise>,<high>,<fall>.
Default values are 1e6,5,0,10,10,10,10

APPLy:RAMP{<freq>,<ampl>,<offs>,<delay>,<rise>,<fall>}

Purpose

This command is a high-level command that programs the PXA125 to output ramp waveform along with its associated parameters

- <freq> sets the output frequency in units of hertz
- <ampl> sets the output amplitude in units of volts
- <offs> sets the output offset in units of volts
- <delay> sets the delay to the pulse rise transition in units of %
- <rise> sets the pulse rise time in units of %
- <fall> sets the pulse fall time in units of %

Parameter type

Numeric

Parameter range

<freq>	10e3 to 50e6
<ampl>	80e-3 to 80
<offs>	-3.6 to +3.6
<delay>	0 to 99.9
<rise>	0 to 99.9
<fall>	0 to 99.9

APPLy:RAMP?

Response

The PXA125 will return <freq>, <ampl>,<offs>,<delay>,<high>,<fall>. Default values are 1e6,5,0,10,10,10

APPLy:SINC{<freq>,<ampl>,<offs>,<N_cycles>}

Purpose

This command is a high-level command that programs the PXA125 to output sinc waveform along with its associated parameters

- <freq> sets the output frequency in units of hertz
- <ampl> sets the output amplitude in units of volts
- <offs> sets the output offset in units of volts
- <N_cycles> sets the number of zero crossing

Parameter type

Numeric

Parameter range

<freq>	10e3 to 50e6
<ampl>	80e-3 to 80
<offs>	-3.6 to +3.6
<N_cycle>	4 to 100

APPLy:SINC?

Response

The PXA125 will return <freq>, <ampl>,<offs>,<N_cycle>. Default values are 1e6,5,0,10

APPLy:EXPonential{<freq>,<ampl>,<offs>,<exp>}

Purpose

This command is a high-level command that programs the PXA125 to output exponential waveform along with its associated parameters

- <freq> sets the output frequency in units of hertz
- <ampl> sets the output amplitude in units of volts
- <offs> sets the output offset in units of volts
- <exp> sets the exponent value

Parameter type

Numeric

Parameter range

<freq>	10e3 to 50e6
<ampl>	80e-3 to 80
<offs>	-3.6 to +3.6
<exp>	-20 to +20

APPLy:EXPonential?

Response

The PXA125 will return <freq>, <ampl>,<offs>,<exp>. Default values are 1e6,5,0,1

APPLy:GAUSsian{<freq>,<ampl>,<offs>,<exp>}

Purpose
This command is a high-level command that programs the PXA125 to output gaussian waveform along with its associated parameters

- <freq> sets the output frequency in units of hertz
- <ampl> sets the output amplitude in units of volts
- <offs> sets the output offset in units of volts
- <exp> sets the exponent value

Parameter type
Numeric

Parameter range

<freq>	10e3 to 50e6
<ampl>	80e-3 to 80
<offs>	-3.6 to +3.6
<exp>	10 to 200

APPLy:GAUSsian?

Response
The PXA125 will return <freq>, <ampl>,<offs>,<exp>. Default values are 1e6,5,0,20

APPLy:DC{<%_ampl>}

Purpose
This command is a high-level command that programs the PXA125 to output DC level along with its associated parameter

- <%_ampl> sets the output level as percentage of the amplitude setting

Parameter type
Numeric

Parameter range

<%_ampl>	-100 to +100 in units of %

APPLy:DC?

Response to query version
The PXA125 will return <%_ampl>. Default values are 100

APPLy:USER{<seg# >,<sclk>,<ampl>,<offs>}

Purpose
This command is a high-level command that programs the PXA125 to output user-defined waveform along with its associated parameters

- <seg# > selects the active segment number
- <sclk> sets the sample clock frequency in units of samples per second
- <ampl> sets the output amplitude in units of volts

- <offs> sets the output offset in units of volts

Parameter type

Numeric

Parameter range

<seg#>	1 to 4096
<sclk>	10e-6 to 100e6
<ampl>	80e-3 to 80
<offs>	-3.6 to +3.6

APPLy:USER?

Response

The PXA125 will return <seg#>,<sclk>, <ampl>,<offs>. Default values are 1,10e6,5,0

FM{OFF|ON|0|1}

Purpose

This command will turn the FM function on and off.

- ON or 1 will set the FM on
- OFF or 0 will set the FM off

Parameter type

Boolean

FM?

Response and default

The PXA125 will return 1 if the FM is on, or 0 if the FM is off. Default is 0.

FM:DEViation<deviation>

Purpose

This command will set the deviation frequency in units of samples per second.

- <deviation> sets the phase offset between the channels. Phase offset is in waveform points

Parameter type

Numeric

Parameter range

<deviation>	1e-3 to 100e6 in units of samples per second

FM:DEViation?

Response and default

The PXA125 will return the present frequency deviation value. The returned value will be in standard scientific format (for example: 100mHz would be returned as 100E-3 – positive numbers are unsigned). Default value is 1e3. Default value is 1e6.

FM:FUNCtion:MODE{FIXed|USER}

Purpose

This command will set the frequency modulating waveform mode to FIXed (standard) and USER (arbitrary) waveforms.

- FIXed will enable a library of built in standard waveforms
- USER will set the generator to accept arbitrary waveform from an external application

Parameter type

Discrete

FM:FUNCtion:MODE?

Response and default

The PXA125 will return FIX, or USER depending on its present setting. Default is FIX.

FM:FUNCtion:SHAPe{SINusoid|TRIangle|SQUare|RAMP}

Purpose

This command will select one of the waveform shapes as the active modulating waveform.

- SINusoid will select sine shape as modulating waveform
- TRIangle will select triangular shape as modulating waveform
- SQUare will select square shape as modulating waveform
- RAMP will select ramp shape as modulating waveform

Parameter type

Discrete

FM:FUNCtion:SHAPe?

Response and default

The PXA125 will return SIN, TRI, SQU, or RAMP depending on its present setting. Default is SIN.

FM:FREQuency<FM_freq>

Purpose

This command will set the modulating wave frequency. This parameter affects the PXA125 when set to operate in FM:FUNC:MODE FIX

- <FM_freq> will set the frequency of the modulating waveform

Parameter type

Numeric

Parameter range

<FM_freq>	1e-3 to 100e6 in units of hertz

FM:FREQuency?

Response and default

The PXA125 will return the present modulating waveform frequency value. The returned value will be in standard scientific format (for example: 100mHz would be returned as 100E-3 – positive numbers are unsigned). Default value is 1e3.

FM:FREQuency:RASTer<FM_sclk>

Purpose

This command will set the modulating wave sample clock frequency. This parameter affects the PXA125 when set to operate in FM:FUNC:MODE USER

- <FM_sclk> will set the sample clock frequency of the modulating waveform

Parameter type

Numeric

Parameter range

<FM_sclk>	1e-3 to 10e6 in units of samples per second

FM:FREQuency:RASTer?

Response and default

The PXA125 will return the present modulating waveform sample clock frequency value. The returned value will be in standard scientific format (for example: 1KHz would be returned as 1E3 – positive numbers are unsigned). Default value is 1e6.

FM:TRIGger:MODE{CONTinuous|TRIGered|GATEd}

Purpose

This command will select one of the FM modes.

- CONTinuous will select continuous modulation
- TRIGered will select triggered modulation
- GATEd will select gated modulation

Parameter type

Discrete

FM:TRIGger:MODE?

Response and default

The PXA125 will return CONT, TRIG, or GATE depending on its present setting. Default is CONT.

FM:TRIGger:SLOPe{POSitive|NEGative}

Purpose

This command will select slope sensitivity for the FM trigger input.

- POSitive will select rising edge
- NEGative will select falling edge

Parameter type

Discrete

FM:TRIGger:SLOPe?

Response and default

The PXA125 will return POS, or NEG depending on its present setting. Default is POS.

FREQuency{<freq>|MINimum|MAXimum}

Purpose

This command modifies the frequency of the standard waveform in units of hertz (Hz). It has no affect on arbitrary waveforms.

- <freq> sets the waveform frequency
- <MINimum> sets the waveform frequency to its lowest value
- <MAXimum> sets the waveform frequency to its highest value

Parameter type

Numeric

Parameter range

<freq>	100e-6 to 50e6

FREQuency?

Response and default

The PXA125 will return the present frequency value. The returned value will be in standard scientific format (for example: 100mHz would be returned as 100E-3 – positive numbers are unsigned). Default value is 1e6.

FREQuency:RASTer{<sclk>|MINimum|MAXimum}

Purpose

This command modifies the sample clock frequency of the arbitrary waveform in units of samples per second (S/s). It has no affect on standard waveforms.

- <sclk> sets the sample clock frequency
- <MINimum> sets the sample clock frequency to its lowest value
- <MAXimum> sets the sample clock frequency to its highest value

Parameter type

Numeric

Parameter range

<sclk>	100e-3 to 100e6

FREQuency:RASTer?

Response and default

The PXA125 will return the present sample clock frequency value. The returned value will be in standard scientific format (for example: 100MHz would be returned as 100E6 – positive numbers are unsigned). Default value is 10e6.

FREQuency:RASTer:SOURce{EXTernal|INTernal}

Purpose

This command selects the source of the sample clock generator. This command affects both the standard and the arbitrary waveforms.

- <EXTernal> selects an external source
- <INTernal> selects the internal source

Parameter type

Discrete

FREQuency:RASTer:SOURce?

Response and default

The PXA125 will return EXT if an external source is selected, or INT if the internal source is selected. Default value is INT.

FSK:FREQuency:RASTer<FM_sclk>

Purpose

This command will set the shift sample clock frequency. When set to operate in FSK mode, the PXA125 will hop from carrier sample clock frequency to shifted sample clock frequency.

- <FSK_sclk>   will set the shifted sample clock frequency for the FSK mode

Parameter type

Numeric

Parameter range

<FSK_sclk>	100e-3 to 100e6 in units of samples per second

FSK:FREQuency:RASTer?

Response and default

The PXA125 will return the present shifted sample clock frequency value. The returned value will be in standard scientific format (for example: 1KHz would be returned as 1E3 – positive numbers are unsigned). Default value is 80e3.

FSK:MODE{HOP|RAMPed}

Purpose

This command will select one of the FSK modes.

- HOP will select non-delayed hops from carrier sclk to shifted sclk
- RAMPed will select ramped shifts from carrier sclk to shifted sclk

Parameter type

Discrete

FSK:MODE?

Response and default

The PXA125 will return HOP, or RAMP depending on its present setting. Default is HOP.

FSK:RAMP:TIME<time>

Purpose

This command will set the time it takes for the sample clock frequency to ramp up and down from carrier to shifted frequencies and back. This parameter will affect the PXA125 only when the instrument is set to operate in ramped FSK mode.

- <time>   will set the ramp up and down time

Parameter type

Numeric

Parameter range

<time>	10e-6 to 1 in units of seconds

FSK:RAMP:TIME?

Response and default

The PXA125 will return the present ramp time value. The returned value will be in standard scientific format (for example: 10ms would be returned as 10E-3 – positive numbers are unsigned). Default value is 1e-3.

FUNCTion:MODE{FIXed|USER|SEQuence}

Purpose

This command defines the type of waveform that will be available at the output connector.

- <FIXed> selects standard waveforms
- <USER> selects arbitrary waveforms
- <SEQuence> selects sequenced waveforms

Parameter type

Discrete

FUNCTion:MODE?

Response and default

The PXA125 will return FIX, USER, or SEQ depending on the present PXA125 setting. Default value is FIX.

FUNCtion:SHAPe{SINusoid|TRIangle|SQUare|PULSe|RAMP|SINC|EXPonential| GAUSsian|NOISe|DC}

Purpose

This command defines the shape of waveform that will be available at the output connector. This command has no affect if present PXA125 setting is USER or SEQ.

- <SINusoid> selects sine waveform
- <TRIangle> selects triangular waveform
- <SQUare> selects square waveform
- <PULSe> selects pulse waveform
- <RAMP> selects ramp waveform
- <SINC> selects sinc waveform
- <EXPonential> selects exponential waveform
- <GAUSsian selects gaussian waveform
- <NOISe> selects random noise waveform
- <DC> selects dc waveform

Parameter type

Discrete

FUNCtion:SHAPe?

Response and default

The PXA125 will return SIN, TRI, SQU, PULS, RAMP, SINC, EXP, GAUS, NOIS, or DC depending on the present PXA125 setting. Default value is SIN.

ROSCillator:SOURce{INTernal|EXTernal|TCXO}

Purpose

This command will select the reference source for the sample clock generator.

- <INTernal> selects CLK10 off the backplane of the PXI chassis
- <EXTernal> activates the external reference input. An external reference must be connected to the PXA125 for it to continue normal operation
- <TCXO> selects an on-board 1ppm TCXO source. Note that the TCXO is an option, so if you did not purchase the instrument with the TCXO option, this command will not be available for you to use.

Parameter type

Discrete

ROSCillator:SOURce?

Response and default

The PXA125 will return INT, EXT, or TCXO depending on the present PXA125 setting. Default value is INT.

SWEep:STOP<stop_sclk>

Purpose

This command will set the stop sample clock frequency. When set to operate in sweep mode, the PXA125 will sweep from carrier sample clock frequency to sweep stop sample clock frequency.

- <stop_sclk> will set the sweep stop sample clock frequency for the sweep mode

Parameter type

Numeric

Parameter range

<stop_sclk>	100e-3 to 100e6 in units of samples per second

SWEep:STOP?

Response and default

The PXA125 will return the present sweep stop sample clock frequency value. The returned value will be in standard scientific format (for example: 1KHz would be returned as 1E3 – positive numbers are unsigned). Default value is 20e6.

SWEep:TIME<time>

Purpose

This command will set the time it takes for the sample clock frequency to sweep from carrier to stop frequencies. This parameter will affect the PXA125 only when the instrument is set to operate in sweep mode.

- <time> will set the sweep time

Parameter type

Numeric

Parameter range

<time>	1e-3 to 1000 in units of seconds

SWEep:TIME?

Response and default

The PXA125 will return the present sweep time value. The returned value will be in standard scientific format (for example: 10ms would be returned as 10E-3 – positive numbers are unsigned). Default value is 1e-3.

SWEep:DIRection{UP|DOWN}

Purpose

This command will select the sweep direction up and down.

- <UP> selects an up direction. The sample clock will sweep from start frequency set by the sclk parameter to stop frequency set by the stop sclk parameter
- <DOWN> selects the down direction. The sample clock will sweep from stop frequency set by the stop sclk parameter to start frequency set by the sclk parameter

Parameter type

Discrete

SWEep:DIRection?

Response and default

The PXA125 will return UP, or DOWN depending on the present PXA125 setting. Default value is UP.

SWEep:SPACing{LINear|LOGarithmic}

Purpose

This command will select the sweep spacing from linear spacing and logarithmic spacing.

- <LINear> will select linear steps
- <LOGarithmic> will select logarithmic steps

Parameter type

Discrete

SWEep:SPACing?

Response and default

The PXA125 will return LIN, or LOG depending on the present PXA125 setting. Default value is LIN.

SWEep:TRIGger:MODE{CONTinuous|TRIGered|GATEd}

Purpose

This command will select one of the sweep modes.

- CONTinuous will select continuous sweep modulation
- TRIGered will select triggered sweep modulation
- GATEd will select gated sweep modulation

Parameter type

Discrete

SWEep:TRIGger:MODE?

Response and default
The PXA125 will return CONT, TRIG, or GATE depending on its present setting. Default is CONT.

SWEep:TRIGger:SLOPe{POSitive|NEGative}

Purpose
This command will select slope sensitivity for the sweep trigger input.

- POSitive will select rising edge
- NEGative will select falling edge

Parameter type
Discrete

SWEep:TRIGger:SLOPe?

Response and default
The PXA125 will return POS, or NEG depending on its present setting. Default is POS.

SWEep:MARKer<mark_sclk>

Purpose
This command will set the marker sample clock frequency for the sweep mode.

- <mark_sclk> will set the marker sample clock frequency

Parameter type
Numeric

Parameter range

<mark_sclk>	100e-3 to 100e6 in units of samples per second

SWEep:MARKer?

Response and default
The PXA125 will return the present sweep sample clock frequency value. The returned value will be in standard scientific format (for example: 1KHz would be returned as 1E3 – positive numbers are unsigned). Default value is 64e6.

VOLTage{<ampl>|MINimum|MAXimum}

Purpose
This command programs the peak-to-peak amplitude of the output waveform. The amplitude is calibrated when the source impedance is 50Ω.

- <ampl> sets the amplitude
- <MINimum> sets the amplitude to its lowest value
- <MAXimum> sets the amplitude to its highest value

Parameter type
Numeric

Parameter range

<ampl>	10e-3 to 10 in units of volts
<MINimum>	10e-3 in units of volts

<MAXimum>	10 in units of volts

VOLTage?

Response to query version

The PXA125 will return the present amplitude value. The returned value will be in standard scientific format (for example: 100mV would be returned as 100E-3 – positive numbers are unsigned). Default value is 5.

VOLTage:OFFSet<offs>

Purpose

This command programs the amplitude offset of the output waveform. The offset is calibrated when the source impedance is 50Ω.

- <offs> sets the amplitude offset

Parameter type

Numeric

Parameter range

<offs>	-3.6 to +3.6 in units of volts

VOLTage:OFFSet?

Response and default

The PXA125 will return the present offset value. The returned value will be in standard scientific format (for example: 100mV would be returned as 100E-3 – positive numbers are unsigned). Default value is 0.

SINusoid:PHASe<phase>

Purpose

This command programs start phase of the standard sine waveform. This command has no affect on arbitrary waveforms.

- <phase> sets the start phase

Parameter type

Numeric

Parameter range

<phase>	0 to 360 in units of degrees

SINusoid:PHASe?

Response and default

The PXA125 will return the present start phase value. Default is 0.

TRIangle:PHASe<phase>

Purpose

This command programs start phase of the standard triangular waveform. This command has no affect on arbitrary waveforms.

- <phase> sets the start phase

Parameter type

Numeric

Parameter range

<phase>	0 to 360 in units of degrees

TRIangle:PHASe?

Response and default

The PXA125 will return the present start phase value. Default value is 0.

SQUare:DCYCle<duty_cycle>

Purpose

This command programs duty cycle of the standard square waveform. This command has no affect on arbitrary waveforms.

- <duty_cycle> sets the duty cycle value

Parameter type

Numeric

Parameter range

<duty_cycle>	0 to 99 in units of percent

SQUare:DCYCle?

Response and default

The PXA125 will return the present duty cycle value. Default value is 50.

PULSe:DELay<delay>

Purpose

This command programs delay of the standard pulse waveform. This command has no affect on arbitrary waveforms.

- <delay> sets the delay parameter

Parameter type

Numeric

Parameter range

<delay>	0 to 99.9 in units of percent

PULSe:DELay?

Response and default

The PXA125 will return the present delay value. Default value is 10.

PULSe:WIDTh<pulse_width>

Purpose

This command programs pulse high portion of the standard pulse waveform. This command has no affect on arbitrary waveforms.

- <pulse_width> sets the high time parameter

Parameter type

Numeric

Parameter range

<pulse_width>	0 to 99.9 in units of percent

PULSe:WIDTh?

Response and default

The PXA125 will return the present width value. Default value is 10.

PULSe:TRANsition<rise>

Purpose

This command programs pulse transition from low to high of the standard pulse waveform. This command has no affect on arbitrary waveforms.

- <rise> sets the rise time parameter

Parameter type

Numeric

Parameter range

<rise>	0 to 99.9 in units of percent

PULSe:TRANsition?

Response and default

The PXA125 will return the present rise time value. Default value is 10.

PULSe:TRANsition:TRAiling<fall>

Purpose

This command programs pulse transition from high to low of the standard pulse waveform. This command has no affect on arbitrary waveforms.

- <fall> sets the fall time parameter

Parameter type

Numeric

Parameter range

<fall>	0 to 99.9 in units of percent

PULSe:TRANsition:TRAiling?

Response and default

The PXA125 will return the present fall time value. Default value is 10.

RAMP:DELay<delay>

Purpose

This command programs delay of the standard ramp waveform. This command has no affect on arbitrary waveforms.

- <delay> sets the delay parameter

Parameter type

Numeric

Parameter range

<delay>	0 to 99.9 in units of percent

RAMP:DELay?

Response and default

The PXA125 will return the present delay value. Default value is 0.

RAMP:TRANsition<rise>

Purpose

This command programs ramp transition from low to high of the standard ramp waveform. This command has no affect on arbitrary waveforms.

- <rise> sets the rise time parameter

Parameter type

Numeric

Parameter range

<rise>	0 to 99.9 in units of percent

RAMP:TRANsition?

Response and default

The PXA125 will return the present rise time value. Default value is 60.

RAMP:TRANsition:TRAiling<fall>

Purpose

This command programs ramp transition from high to low of the standard ramp waveform. This command has no affect on arbitrary waveforms.

- <fall> sets the fall time parameter

Parameter type

Numeric

Parameter range

<fall>	0 to 99.9 in units of percent

RAMP:TRANsition:TRAiling?

Response and default

The PXA125 will return the present fall time value. Default value is 30.

SINC:NCYCleN_cycles>

Purpose

This command programs the number of “0-crossings” of the standard SINC pulse waveform. This command has no affect on arbitrary waveforms.

- <N_cycle> sets the number of zero-crossings parameter

Parameter type

Numeric (integer only)

Parameter range

<N_cycle>	4 to 100 zero crossings

SINC:NCYCle?

Response and default

The PXA125 will return the present number of zero-crossing value. Default value is 10.

GAUSsian:EXPonent<exp>

Purpose

This command programs the exponent for the standard gaussian pulse waveform. This command has no affect on arbitrary waveforms.

- <exp > sets the exponent parameter

Parameter type

Numeric

Parameter range

<exp>	10 to 200

GAUSsian:EXPonent?

Response and default

The PXA125 will return the present exponent value. Default value is 20.

EXPonential:EXPonent<exp>

Purpose

This command programs the exponent for the standard exponential waveform. This command has no affect on arbitrary waveforms.

- <exp > sets the exponent parameter

Parameter type

Numeric

Parameter range

<exp>	-100 to 100

EXPonential:EXPonent?

Response and default

The PXA125 will return the present exponent value. Default value is –20.

DC<%_amplitude>

Purpose

This command programs the amplitude of the standard DC waveform. This command has no affect on arbitrary waveforms.

- <%_amplitude > sets the dc level as percentage of the programmed amplitude setting

Parameter type

Numeric

Parameter range
<%_amplitude> -100 to 100 in units of percent

DC?

Response and default
The PXA125 will return the present dc value. Default value is 100.

OUTPut Subsystem

This subsystem controls the characteristics of the output. It controls filter type and filter cutoff frequency. It is also used to control the sync output, source and position. Optional nodes were omitted from these commands. Factory defaults after *RST are shown in bold typeface. Parameter low and high limits are given where applicable.

Keyword Limit	Parameter Form	Default,Low Limit,High
:OUTPut (?)	{OFF|ON|0|1}	**OFF**
:FILTer (?)	{NONE|25M|50M|ALL}	**NONE**
:SYNC (?)	{OFF|ON|0|1}	**OFF**
:SOURce (?)	{BIT|LCOMplete}	**BIT**
:POSition (?)	<position>	**0**,0,n
:WIDTh (?)	<width>	**4**,4,100000
:TTLTrig<N>	{1|2|3|4|5|6|7}	
[:STATe] (?)	{OFF|ON|0|1}	**OFF**
:STAR	{1|2|3|4|5|6|7}	
[:STATe] (?)	{OFF|ON|0|1}	**OFF**

OUTPut{OFF|ON|0|1}

Purpose
This command will turn the PXA125 output on and off.

- ON or 1 will set the output on
- OFF or 0 will set the output off

Parameter type
Discrete

OUTPut?

Response and default
The PXA125 will return 1 if the output is on, or 0 if the output is off. Default value is 0.

OUTPut:FILTer{NONE|25M|50M|ALL}

Purpose
This command will select which filter is connected to the PXA125 output. Observe the following restrictions when you try to use this command:

1) Filter selection is not available when the instrument is set to output the standard sine waveform. In fact, the default waveform shape is sine. Therefore, filter selection will be available for use only after you select a different waveform, or change the output mode to use.
2) Filters are placed before the output amplifier. Therefore, do not expect the filters to remove in-band amplifier harmonics and spurious.

- NONE will disconnect all filters from the output path
- 25M will connect 25 MHz low pass elliptic filter
- 50M will connect 50 MHz low pass elliptic filter
- ALL will connect both the 25 MHz and the 50 MHz filters in series

Parameter type
Discrete

OUTPut:FILTer?

Response and default
The PXA125 will return NONE, 25M, 50M, or ALL depending on the type of filter presently connected to the output. Default value is NONE.

OUTPut:SYNC{OFF|ON|0|1}

Purpose
This command will turn the PXA125 SYNC output on and off.

- ON or 1 will set the SYNC output on
- OFF or 0 will set the SYNC output off

Parameter type
Discrete

OUTPut:SYNC?

Response and default
The PXA125 will return 1 if the SYNC output is on, or 0 if the SYNC output is off. Default value is 0.

OUTPut:SYNC:SOURce{BIT|LCOMplete}

Purpose
This command will select the PXA125 SYNC option.

- BIT will set the SYNC to output a narrow pulse
- LCOMplete will set the SYNC to transition high at the beginning of the sequence and low a the end of the sequence

Parameter type
Discrete

OUTPut:SYNC:SOURce?

Response and default
The PXA125 will return BIT, or LCOM depending on the present setting of the SYNC source. Default value is BIT.

OUTPut:SYNC:POSition<position>

Purpose
This command will program the PXA125 SYNC position. This command is active in arbitrary (USER) mode only.

- <position> will set the SYNC position in waveform points

Parameter type
Numeric (integer only)

Parameter range

<position>	0 to “n” in units of waveform points, where “n” is the size of the active segment. The sync position can be programmed in increments of 4 points minimum.

OUTPut:SYNC:POSition?

Response and default
The PXA125 will return the present duty cycle value. Default value is 50.

Response to query version
The PXA125 will return the present SYNC position value. Default value is 0.

OUTPut:SYNC:WIDTh<width>

Purpose
This command will program the width of the SYNC pulse. The sync width can be varied through a limited range to allow widening of the sync pulse.

- <width> will set the SYNC width

Parameter type
Numeric (integer only)

Parameter range

<width>	4 to 100000 in units of waveform points.

OUTPut:SYNC:WIDTh?

Response and default
The PXA125 will return the present SYNC width value. Default value is 4.

OUTPut:TTLTrig<N>{OFF|ON|0|1}

Purpose
The backplane on the PXI chassis has some signals on the P2 connector that allow inter-instrument communication and synchronization. Among these lines, there are some trigger lines, STAR trigger, and local bus lines. There are 8 trigger lines that can serve as source or target. Normally, these lines are designated as tri-state lines. The OUTP:TTLT<N> command connects a trigger pulse to one or more of the backplane trigger lines. <N> can range from 0 through 7. The trigger pulse is an exact duplication of the SYNC output pulse. Therefore, programming the width and location of the SYNC output, has the same affect on the TTLTrig lines. The outp:ttlt command can be applied to 1 or more lines.

- <N> can be selected from 0 through 7
- ON or 1 will convert the designated TTLT line to an output and will place a SYNC pulse on this line
- OFF or 0 will set the designated TTLT line output off, the Sync pulse removed and line converted to high-impedance, tri-state.

Parameter type
Discrete

OUTPut:TTLTrig<N>?

Response and default

The PXA125 will return the present designated TTLT line state: 1 if the SYNC output is on, or 0 if the SYNC output is off. Default value is 0.

OUTPut:STAR{OFF|ON|0|1}

Purpose

The STAR pin at the P2 is normally used as an input. STAR triggers are sourced from a STAR module that is plugged in slot 2 however, by making the STAR pin output, the instrument can send trigger information that can be interpreted as status information by the STAR module.

- ON or 1 will convert the STAR line to output and will place a SYNC pulse on this line
- OFF or 0 will set the STAR line output off, the Sync pulse removed and the line converted to high-impedance, tri-state.

Parameter type

Discrete

OUTPut:STAR?

Response and default

The PXA125 will return the STAR line state: 1 if the STAR is on (output), or 0 if the STAR is off (input). Default value is 0.

INSTrument Subsystem

A dual channel arbitrary waveform generator is considered as two logical instruments. The INSTrument subsystem provides a mechanism to identify and select instrument channels. Factory defaults after *RST are shown in bold typeface. Parameter low and high limits are given where applicable.

Keyword	Parameter Form	Default,Low Limit,High Limit
:INSTrument (?)	{1|2}	**1**
:COUPle (?)	{OFF|ON|0|1}	**OFF**
:MODE (?)	(MASTer|SLAVe)	**MASTer**
:PHASe (?)	<phase>	**0**,0,n
:SET	{0|1|2}	**0**,0,2

INSTrument:COUPle{OFF|ON|0|1}

Purpose

This command will enable daisy-chained instruments to operate in multi-instrument synchronization mode.

- ON or 1 will set the couple mode on
- OFF or 0 will set the couple mode off

Parameter type

Discrete

INSTrument:COUPle?

Response and default

The PXA125 will return 0, or 1 depending on the present instrument couple setting. Default value is 0.

INSTrument:COUPle:MODE{MASTer|SLAVe}

Purpose

This command will select master and slave instruments in multi-instrument synchronization mode. Note that only one instrument in a chain may be set as master. All other instruments must be set as slaves.

- MASTer will set the master instrument
- SLAVe will set the slave instrument

Parameter type

Discrete

INSTrument:COUPle:MODE?

Response and default

The PXA125 will return MAST, or SLAV depending on the present instrument couple setting. Default value is MAST.

INSTrument:COUPle:PHASe<phase>

Purpose

This command will program the phase offset between master and slave units. This command has no effect on the instrument in modes other than instrument couple.

- <phase> will set the leading edge offset between master and slave units

Parameter type

Numeric (integer only)

Parameter range

<phase>	0 to “n” in units of waveform points, where “n” is the size of the active segment. The phase offset can be programmed in increments of 4 points.

INSTrument:COUPle:PHASe?

Response to query version

The PXA125 will return the present phase offset value. Default value is 0.

INSTrument:SET {0|1|2}

Purpose

This command will prepare the master and slave units for multi-instrument synchronization. The procedure how to synchronize multiple instruments is given below.

Parameter type

Numeric (integer only)

Synchronizing Multiple Instruments

The capability to synchronize multiple instruments is one of the strongest features of the Model PXA125. However, since each instrument is free running and has completely independent settings and controls, one must first prepare the instruments to a level where all instruments can unite.

There are some simple steps to synchronize instruments however, three conditions must be met otherwise synchronization will fail, they are:

1. **Synchronized cards must be adjacent to each other, and**
2. **Master module must be at the left of the clustered instruments**
3. **The module, which is at the far right side of the cluster, is designate as "Last slave".**

NOTE

Every cluster of synchronized instruments must have one master and one last slave. The master module is always at the left side of the cluster and the last slave is always at the far right side of the cluster. Any number from 0 to n modules can be placed between the master and the slave modules.

After you have your master and slave units in place, you'll have to execute a series of commands to prepare the instruments to their final lock position. The following example assumes three adjacent instruments. As was explained before, the left instrument is called "master", the center instrument is called "slave" and the instrument on the right is called "last slave". Here is what you do:

1. Program "slave": inst:coup:mode slav
2. Program "last slave": inst:coup:mode slav
3. Program "master": inst:coup on
4. Program "slave": inst:coup on
5. Program "last slave": inst:coup on
6. Program "last slave": inst:coup:set 0
7. Program "slave": inst:coup:set 1
8. Program "master": inst:coup:set 1
9. Program "last slave": inst:coup:set 2

After you send the last command, all instruments will be locked together to the sample clock frequency and start phase of the master signal.

You can still program each module with some commands and retain lock position however, some operations require that you repeat the lock-in sequence as shown above. In the following you'll find a list of operation that will not interfere with synchronization:

1. Changing output state on/off
2. Changing sync output state on/off
3. Changing sync validation bit/lcom
4. Modifying amplitude and offset
5. Modifying sample clock frequency

In the following you'll find a list of operation that will interfere with synchronization:

1. Changing waveform type fix/user/sequence
2. Downloading waveforms
3. Modifying sync position and width
4. Modifying trigger burst count, phase and advance source
5. Modifying phase offset between instruments
6. Modifying frequency setting in fixed waveforms mode

TRIGger Subsystem

The TRIGger subsystem is used to synchronize device actions with external events. These commands control the trigger modes of the Model PXA125. The generator can be placed in Triggered, Gated or Burst mode. Trigger source is selectable from an external source, an internal trigger generator or a software trigger. Optional nodes were omitted from these commands. Factory defaults after *RST are shown in bold typeface. Parameter low and high limits are given where applicable.

Keyword	**Parameter Form**	**Default,**Low Limit,High Limit
:ARM (?)	{OFF|ON|0|1}	**ON**
:SLOPe	(POSitive|NEGative)	**POSitive**
:BREakpoint		
:POSition (?)	<position>	**0**,0,n
:INITiate		
:CONTinuous (?)	{OFF|ON|0|1}	**ON**
:TRIGger		
:BURSt (?)	{OFF|ON|0|1}	**OFF**
:COUNt ?)	<count>	**1**,1,128K
:GATE (?)	{OFF|ON|0|1}	**OFF**
:PHASe (?)	<phase>	**0**,0,n
:SLOPe (?)	{POSitive|NEGative}	**POSitive**
:SOURce		
:ADVance (?)	{EXTernal|INTernal|TTLTrig0-7|STAR}	**EXTernal**
:STAR (?)	{OFF|ON|0|1}	**OFF**
:TIMer (?)	<interval>	**1e3**,100e-3,5e6
:IMMediate		
*TRG		

ARM{OFF|ON|0|1}

Purpose

This command will arm the PXA125 to stop and start generating output waveforms. The instrument will stop at a breakpoint set with one of the arm command.

- ON or 1 will set the arm mode on
- OFF or 0 will set the arm mode off

Parameter type

Discrete

ARM?

Response and default

The PXA125 will return 1 if the arm mode is on, or 0 if the arm mode is off. Default value is 0.

ARM:SLOPe{POSitive|NEGative}

Purpose

This command will set the edge sensitivity of which the PXA125 will stop in arm mode. The stop and start commands are applied at the rear-panel TRIG IN connector.

- POSitive will set the positive edge
- NEGative will set the negative edge

Parameter type

Discrete

ARM:SLOPe?

Response and default

The PXA125 will return POS, or NEG depending on the present setting. Default value is POS.

ARM:BREakpoint:POSition<position>

Purpose

This command will program the breakpoint position for the arm function.

- <position> will set the breakpoint position

Parameter type

Numeric (integer only)

Parameter range

<position>	0 to “n” in units of waveform points, where “n” is the size of the active segment. The breakpoint position can be programmed in increments of 4 points minimum.

ARM:BREakpoint:POSition?

Response and default

The PXA125 will return the present breakpoint value. Default value is 0.

INITiate:CONTinuous{OFF|ON|0|1}

Purpose

This command will set the output in continuous operation and interrupted operation. The trigger subsystem commands will affect the PXA125 only after it will be set to interrupted operation.

- ON or 1 will set the continuous mode
- OFF or 0 will set the interrupted mode

Parameter type

Discrete

INITiate:CONTinuous?

Response to query version

The PXA125 will return 1 if the output is continuous, or 0 if the output is interrupted. Default value is 0.

TRIGger:BURSt{OFF|ON|0|1}

Purpose

This command will set the burst mode. This command will affect the PXA125 only after it will be set to interrupted operation with the command INIT:CONT OFF.

- ON or 1 will set the burst mode on
- OFF or 0 will set the burst mode off

Parameter type

Discrete

TRIGger:BURSt?

Response and default

The PXA125 will return 1 if the burst is on, or 0 if the burst is off. Default value is 0.

TRIGger:COUNt<count>

Purpose

This command will set the burst counter.

- <count> will set the count number

Parameter type

Numeric (integer only)

Parameter range

<count>	1 to 128K

TRIGger:COUNt?

Response and default

The PXA125 will return the present count value. Default value is 1.

TRIGger:GATE{OFF|ON|0|1}

Purpose

This command will set the gate mode. This command will affect the PXA125 only after it will be set to interrupted operation with the command INIT:CONT OFF.

- ON or 1 will set the gate mode on
- OFF or 0 will set the gate mode off

Parameter type

Discrete

TRIGger:GATE?

Response and default

The PXA125 will return 1 if the gate is on, or 0 if the gate is off. Default value is 0.

TRIGger:SLOPe{POSitive|NEGative}

Purpose
This command will set the edge sensitivity for the trigger input.

- <POSitive> will set the positive edge
- <NEGative> will set the negative edge

Parameter type
Discrete

TRIGger:SLOPe?

Response and default
The PXA125 will return POS, or NEG depending on the present trigger setting. Default value is POS.

TRIGger:SOURce:ADVance{EXTernal|INTernal|TTLTrig<N>|STAR}

Purpose
This command will set the advance source for the trigger mode. The backplane on the PXI chassis has some signals on the P2 connector that allow inter-instrument communication and synchronization. Among these lines, there are some trigger lines, STAR trigger, and local bus lines. There are 8 trigger lines that can serve as source or target. Normally, these lines are designated as tri-state lines. The TRIG:SOUR:TTLT<N> command converts one of these lines to input. <N> can range from 0 through 7. Note that only one input can be selected as an advance source.

- <EXTernal> will select the external input
- <INTernal > will select the internal trigger generator
- <N> can be selected from 0 through 7
- <STAR> will select the STAR trigger module. Select this option only if you have a STAR module in your system

Parameter type
Discrete

TRIGger:SOURce:ADVance?

Response and default
The PXA125 will return EXT, INT, TTLT<N>, or STAR depending on the present trigger advance source setting. Default value is EXT.

TRIGger:STAR{OFF|ON|0|1}

Purpose
In addition to bused PXI triggers, the PXI bus has included an independent trigger (PXI_STAR) for each slot that is oriented in a star configuration from the star trigger slot. The star trigger slot is adjacent to the system slot and uses the 13 left local bus signals as the star triggers. The PXI_STAR specifications are only partially implemented in the PXA125 where only 8 star lines are being addressed, all of them in parallel thus, if you plug the card in slot 2 and turn TRIG:STAR ON, the front-panel SYNC signal will be routed to 8 STAR lines simultaneously. The PXA125 controls star lines in slots 3 through 10.

- ON or 1 will make STAR lines in slots 3 through 10 active

- OFF or 0 will turn off STAR triggers

Parameter type

Discrete

TRIGger:STAR?

Response and default

The PXA125 will return 1 if star lines are active, or 0 if star lines are disabled. Default value is 0.

TRIGger:TIMer<interval>

Purpose

This command will set the period for the internal trigger generator.

- <interval> will set the timer of the internal trigger generator

Parameter type

Numeric

Parameter range

<interval>	100e-3 to 5e6 in units of seconds

TRIGger:TIMer?

Response and default

The PXA125 will return the present internal trigger timer value. The returned value will be in standard scientific format (for example: 10ms would be returned as 10E-3 – positive numbers are unsigned). Default value is 1e3.

TRIGger:IMMediate

Purpose

This command will stimulate the PXA125 output once. This command will affect the PXA125 only after it will be set to interrupted operation.

*TRG

Purpose

This common command has the same affect as the TRIG:IMM command.

TRACe Subsystem

The TRACe subsystem contains commands, which allow definition of segments and their corresponding length, addition and deletion of segments, and the loading of waveform data. Sequence commands control segments links and loops. Description of the SendBlock() command is also given. This command places PXA125 in a special data transfer mode where the generator's message-based interface is bypassed and data is loaded directly from the data bus. Optional nodes were omitted from these commands. Defaults are shown in bold. General description of arbitrary waveforms is also included in this section.

Keyword	**Parameter Form**	**Default,**Low Limit,High Limit
:TRACe		
:DEFine	<segment_number>,<length>	
:DELete	<segment_number>	
:DELete:ALL		
:SELect (?)	<segment_number>	
:SEQuence	#<header><binary_block>	
:ADVance (?)	{AUTOmatic|STEP|SINGle|MIXed}	**AUTOmatic**
:SOURce (?)	{EXTernal|INTernal|TTLTrig0-7|STAR}	**EXTernal**
:DEFine	<link>,<seg_#>,<loop>,<mode>	
:DELete:ALL		

Arbitrary Waveforms

Arbitrary waveforms are generated from digital data points, which are stored in memory. Each data point has a vertical resolution of 14 bits (16384 points), i.e., each sample is placed on the vertical axis with a precision of 1/16384. The Model PXA125 has the following waveform memory capacity:

2 Meg – standard memory configuration

Each horizontal point has a unique address - the first being 00000 and the last depends on the memory option. In cases where smaller waveform lengths are required, the waveform memory can be divided into smaller segments.

When the instrument is programmed to output arbitrary waveforms, the clock samples the data points (one at a time) from address 0 to the last address. The rate at which each sample is replayed is defined by the sample clock rate parameter. The PXA125 provides programmable sample clock rates from 100 mS/s to 100 MS/s.

Unlike the built-in standard waveforms, arbitrary waveforms must first be loaded into the instrument's memory. Correct memory management is required for best utilization of the arbitrary memory. An explanation of how to manage the arbitrary waveform memory is given in the following paragraphs.

Arbitrary Memory Management

The arbitrary memory in comprised of finite length of points. The maximum size arbitrary waveform that can be loaded into memory depends on the option that is installed in your instrument. The various options are listed in Chapter 1 of this manual. If you purchased the PXA125 with in its basic configuration, you should expect to have 2 Meg words to load waveforms.

Waveforms are created using small sections of the arbitrary memory. The memory can be partitioned into smaller segments (up to 4096) and different waveforms can be loaded into each segment, each having a unique length. Minimum segment size is 16 points. Information on how to partition the memory, define segment length and download waveform data to the PXA125 is given in the following paragraphs.

Downloading Arbitrary Waveforms

There are two ways to download arbitrary waveform data points into the memory. You can use Wave Composer utility in ArbConnection to program segments and their size, create the waveform on the screen and then with a click of a button, download the entire waveform to the instrument. On the other hand, if you are writing your own application, you'll need to use the SendComand() and SendBlock() functions. Information on these functions is given at the beginning of this chapter. You can follow the general guidelines below to download waveforms to the instrument. The following example assumes segment 1 having 1000 points.

1. Use the function SendCommand() to program trac:sel 1. This command will make segment number 1 active. Anything that you will download to the PXA125 will be downloaded to the active segment, in this case – segment 1.

2. Use the SendCommand() to program trac:def 1,1000. This command defines the size of segment 1, in this case – 1000 waveform points.

3. Use the function SendBlock() to send waveform data points to the PXA125. You can use this function in two ways: 1) You can generate and save an external file with the waveform coordinates and then call this function with the path to your file, or 2) you can build the coordinates into your function, as described earlier in this chapter.

After you download the waveform coordinates to the instrument, you'll have to change the waveform mode to arbitrary (USER) and set the sample clock frequency to that the output frequency will meet with your requirements. Additional information on the commands used for arbitrary waveform management is given below.

HINT

If you use multiple segments in your application, you can combine all segments to one long waveform and then use trac:def n,m commands to define segments length.

TRACe:DEFine<segment_number>,<length>

Purpose

This command will define waveform segments and their relative length. The use of this command is absolutely a must before you can download data using the SendBlock() function.

- <segment_number> will set the selected segment
- <length> will assign length to the above selected segment. Minimum segment length is 16 points, the maximum is limited by the memory option that is installed in your instrument either 1Meg or 4Meg.

NOTE

The PXA125 operates in interlaced mode where four memory cells generate one byte of data. Therefore, segment size can be programmed in numbers evenly divisible by four only. For example, 2096 bytes is an acceptable length for a binary block. 2002 is not a multiple of 4, therefore the generator will generate an error message if this segment length is used.

TRACe:DELete<segment_number>

Purpose

This command will delete a segment. The memory space that is being freed will be available for new waveforms as long as the new waveform is equal or smaller in size then the deleted segment. If the deleted segment is the last segment, then the size of another waveform written to the same segment is not limited. For example, let consider two segments, the first being a 1000-point waveform and the second with 100 points. If you delete segment 1, you can reprogram another waveform to segment 1 with size to 1000 points. If you reprogram segment 1 with 1004 points, the instrument will generate an error and will not accept this waveform. On the other hand, if you delete segment 2, which was the last segment you programmed, then you can reprogram this segment with waveforms having length limited only by the size of the entire memory space.

- <segment_number> will select the segment number that will be deleted

TRACe:DELete:ALL

Purpose

This command will delete all segments and will clear the entire waveform memory. This command is particularly important in case you want to de-fragment the entire waveform memory and start building your waveform segments from scratch.

TIP

The TRAC:DEL:ALL command does not re-write the memory so, whatever waveforms were downloaded to the memory are still there for recovery. The TRAC:DEL:ALL

command removes all stop bits and clears the segment table. You can recover memory segments by using the TRAC:DEF command. You can also use this technique to resize, or combine waveform segments.

TRACe:SELect<segment_number>

Purpose
This command will select the active waveform segment for the output. By selecting the active segment you are performing two functions:

1. Successive :TRAC commands will affect the selected segment
2. The SYNC output will be assigned to the selected segment. This behavior is especially important for sequence operation, where multiple segments form a large sequence. In this case, you can synchronize external devices exactly to the segment of interest

- <segment_number> will set the active waveform segment number

Parameter type
Numeric (integer only)

Parameter range

<segment_number>	1 to 4096

TRACe:SELect?

Response
The PXA125 will return the active segment number.

Sequenced Waveforms

Sequenced waveforms are made of a number of arbitrary waveforms, which can be linked and looped in user-programmable order. Sequenced waveforms are generated from waveforms stored in the PXA125 as memory segments. Therefore, before a sequence can be used, download waveform segments to the arbitrary memory using TRAC# or DMA methods. Information on how to partition the memory and how to download waveforms is given in the section entitled Downloading Arbitrary Waveforms.

An example of how sequenced waveforms work is demonstrated in figure 1-1a through 1-6d. The sequence generator lets you link and loop segments in user-defined order. Figure 1-6d shows a sequence of waveforms that were stored in three different memory segments.

There are a number of tools that you can use to build a sequence table. The easiest way is of course to use the ArbExplorer program. Information how to use the ArbExplorer program is given in a later chapter. In other cases, SCPI programming allows low-level programming of sequence tables.

Use the information below to understand sequence commands and how to implement them in your application.

SEQuence:ADVance{AUTOmatic|STEP|SINGle|MIXed}

Purpose

This command will select the sequence advance mode. The way the instrument advances through the sequence links can be specified by the user.

- **AUTOmatic** - specifies continuous advance where the generator steps continuously to the end of the sequence table and repeats the sequence from the start. For example, if a sequence is made of three segments 1, 2 and 3, the sequence will generate an infinite number of 1,2,3,1,2,3,1,2,3…waveforms. Of course, each link (segment) can be programmed with its associated loop (repeat) number. AUTO is the default sequence advance mode.

- In **STEP** advance mode, the sequence is advanced to the next waveform only when a valid trigger is received. The output of the PXA125 generates the first segment continuously until a trigger signal advances the sequence to the next segment. If repeats were selected for a segment, the loop counter is executed automatically.

- In **SINGle** advance mode, the generator idles between steps until a valid trigger signal is sensed. This mode operates with trigger mode only. An attempt to select the SING advance mode when the PXA125 is in continuous operating mode will generate an error. After trigger, the generator outputs one waveform cycle. Then, the output level idles at a DC level equal to the last point of the last generated waveform. If loops (repeats) were programmed, the output will repeat this segment every time a trigger is received. Only after executing all of the programmed loops will the sequence step to the next assigned segment.

- **MIXed** mode is a special mode, which combines continuous step advance with single step advance in a sequence. There are three conditions for the sequence generator to operate in this mode:

1) The PXA125 must be set to operate in continuous mode
2) Select the MIX sequence advance mode
3) Assign the mixed mode bits for each sequence step in your SEQ:DEF command. “0” programs normal advance, “1” programs trigger advance. Step with a “0” bit assigned to it will advance automatically to the next step. If “1” is assigned to a step, the instrument will generate this step and its associated number of repeats continuously and only a valid trigger signal will advance this step to the next step.

SEQuence:ADVance?

Response and default

The PXA125 will return the AUTO, STEP, SING, or MIX depending on the present sequence advance mode setting. Default value is AUTO

SEQuence:ADVance:SOURce{EXTernal|INTernal|TTLTrig<N>|STAR}

Purpose

This command will select the sequence advance source. The source of the event causing sequence advance can be specified by the user. The backplane on the PXI chassis has some signals on the P2 connector that allow inter-instrument communication and synchronization. Among these lines, there are some trigger lines, STAR trigger, and local bus lines. There are 8 trigger lines that can serve as source or target. Normally, these lines are designated as tri-state lines. The TRIG:SOUR:TTLT<N> command converts one of these lines to input. <N> can range from 0 through 7. But only one line can be selected as an advance source.

- <EXTernal> will set the external advance mode
- <INTernal> will set the internal advance mode
- <N> can be selected from 0 through 7

- <STAR> will select the STAR trigger module. Select this option only if you have a STAR module in your system

SEQuence:ADVance:SOURce?

Response and default

The PXA125 will return EXT, INT, TTLT<N>, or STAR depending on the present trigger advance source setting. Default value is EXT.

SEQuence:DEFine <step_#>,<segment_#>,<#_repeat>,<mode>

Purpose

This command will define sequence steps and their associated advance mode.

- <step_#> will define the link number
- <segment_#> will select the segment for the above link
- <#_repeat> will define how many times the selected segment will loop
- <mode> will assign the mixed mode bits for this sequence step. “0” programs normal advance, “1” programs trigger advance. Step with a “0” bit assigned to it will advance automatically to the next step. If “1” is assigned to a step, the instrument will generate this step and its associated number of repeats continuously and only a valid trigger signal will advance this step to the next step. Note that the <mode> parameter has no affect when the sequence advance mode is set to SING.

To understand better the way the <mode> switch affects a sequence, consider the following sequence setting:

```
 SEQ:DEF 1,1,1,0         (Step 1 = segment 1, no loops, mode 0)
SEQ:DEF 2,2,1,1          (Step 2 = segment 2, no loops, mode 1)
SEQ:DEF 3,3,10,1         (Step 3 = segment 3, 10 loops, mode 1)
SEQ:DEF 4,4,1,0          (Step 4 = segment 4, no loops, mode 0)
```

After selecting the sequence mode, the generator will generate segment 1 and automatically advance to segment 2. It will continue to generate segment 2 until a valid trigger is received. After it has received a trigger, the generator will complete segment 2 and advance to segment 3. It will continue to generate and loop segment 2 until a valid trigger is received. After it has received a trigger, the generator will complete 10 loops of segment 3 and will advance to segment 4. From segment 4 the generator will automatically advance to the beginning of the sequence and will repeat the above.

NOTE

Although trigger signals are used to advance mixed mode, the mixed mode operates in continuous mode only. The <mode> parameter will be ignored if you will use SING as advance mode for the sequence table.

TIP

Every time you use the SEQ:DEF command while your PXA125 is in sequenced operating mode, the instrument attempts to rebuild the sequence table and restart the sequence. Therefore, sending this command in sequenced mode will slow the programming process and the operation of the generator. Using the SEQ:DEF command in FIX or USER mode will greatly speed up programming time.

SEQuence:DELete:ALL

Purpose

This command will erase the sequence table.

SYSTem Subsystem

The system-related commands are not related directly to waveform generation but are an important part of operating the PXA125. These commands can reset or test the instrument, or query the instrument for system information.

Keyword **Parameter Form** **Default,**Low Limit,High Limit

```
:SYSTem
   :ERRor?
   :VERSion?
   :IDENtification?
RESet
*RST
*IDN?
```

SYSTem:ERRor?

Purpose
This query will interrogate the PXA125 for programming errors.

Response to query
The PXA125 will return error code. Error messages are listed later in this manual.

SYSTem:VERSion?

Purpose
This query will interrogate the PXA125 for SCPI version.

Response to query
The PXA125 will return 1999.0.

SYSTem:IDENtification?

Purpose
This query will return instrument identification.

Response to query
The PXA125 will return a string similar to this one: 0x1570, 0x500, 0000000278. The first field is the vendor ID, the second is the device ID and the last field the serial number of the instrument.

RESet, *RST

Purpose
This command will reset the PXA125 to factory defaults.

*IDN?

Purpose
This query will interrogate the identity of the PXA125.

Response to query
The PXA125 will return LeCroy,PXA125,0,1.0. The first field is the instrument model, the second is always “0”, the third is for the firmware version number and the last for instrument options.

IEEE-STD-488.2 Common Commands and Queries

Since most instruments and devices in an ATE system use similar commands that perform similar functions, the IEEE-STD-488.2 document has specified a common set of commands and queries that all compatible devices must use. This avoids situations where devices from various manufacturers use different sets of commands to enable functions and report status. The IEEE-STD-488.2 treats common commands and queries as device dependent commands. For example, *TRG is sent over the bus to trigger the instrument. Some common commands and queries are optional, but most of them are mandatory.

The following is a complete listing of all common-commands and queries, which are used by the PXA125

***CLS** - Clear the Status Byte summary register and all event registers.

***ESE** <enable_value> - Enable bits in the Standard Event enable register. The selected bits are then reported to the status byte.

***ESE?** - Query the Standard Event enable register. The generator returns a decimal value, which corresponds to the binary-weighted sum of all bits, set in the register.

***ESR?** - Query the Standard Event register. The generator returns a decimal value, which corresponds to the binary-weighted sum of all bits, set in the register.

***IDN?** - Query the generator's identity. The returned data is organized into four fields, separated by commas. The generator responds with its manufacturer and model number in the first two fields, and may also report its serial number and options in fields three and four. If the latter information is not available, the device must return an ASCII 0 for each. For example, Model PXA125 response to *IDN? is:

LeCroy,PXA125,0,1.0

***OPC** - Set the "operation complete" bit (bit 0) in the Standard Event register after the previous commands have been executed.

***OPC?** - Returns "1" to the output buffer after all the previous commands have been executed. *OPC? is used for synchronization between a controller and the instrument using the MAV bit in the Status Byte or a read of the Output Queue. The *OPC? query does not affect the OPC Event bit in the Standard Event Status Register (ESR). Reading the response to the ***OPC?** query has the advantage of removing the complication of dealing with service requests and multiple polls to the instrument. However, both the system bus and the controller handshake are in a temporary hold-off state while the controller is waiting to read the *OPC? query response.

***OPT?** - Returns the value "0" for a PXA125 with no options. Returns "1" for a Model PXA125 with 2Meg memory and returns "2" for a Model

PXA125 with TCXO option.

***RST** - Resets the generator to its default state. Default values are listed in Table 4-1.

***SRE** <enable_value> - Enables bits in the Status Byte enable register.

***SRE?** - Query the Status Byte enable register. The generator returns a decimal value in the range of 0 to 63 or 128 to 191 since bit 6 (RSQ) cannot be set. The binary-weighted sum of the number represents the value of the bits of the Service Request enable register.

***STB?** - Query the Status Byte summary register. The *STB? command is similar to a serial poll but is processed like any other instrument command. The *STB? command returns the same result as a serial poll, but the "request service" bit (bit 6) is not cleared if a serial poll has occurred.

***TRG** - Triggers the generator from the remote interface. This command effects the generator if it is first placed in the Trigger or Burst mode of operation and the trigger source is set to "BUS".

The SCPI Status Registers

The Model PXA125 uses the Status Byte register group and the Standard Event register group to record various instrument conditions. Figure 4-1 shows the SCPI status system.

An Event Register is a read-only register that reports defined conditions within the generator. Bits in an event register are latched. When an event bit is set, subsequent state changes are ignored. Bits in an event register are automatically cleared by a query of that register or by sending the *CLS command. The *RST command or device clear does not clear bits in an event register. Querying an event register returns a decimal value, which corresponds to the binary-weighted sum of all bits, set in the register.

An Event Register defines which bits in the corresponding event register are logically ORed together to form a single summary bit. The user can read from and write to an Enable Register. Querying an Enable Register will not clear it. The *CLS command does not clear Enable Registers but it does clear bits in the event registers. To enable bits in an enable register, write a decimal value that corresponds to the binary-weighted sum of the bits required to enable in the register.

The Status Byte Register (STB)

The Status Byte summary register contains conditions from the other registers. Query data waiting in the generator's output buffer is immediately reported through the Message Available bit (bit 4). Bits in the summary register are not latched. Clearing an event register will clear the corresponding bits in the Status Byte summary register. Description of the various bits within the Status Byte summary register is given in the following:

Bit 0 - Decimal value 1. Not used, always set to 0.

Bit 1 - Decimal value 2. Not used, always set to 0.

Bit 2 - Decimal value 4. Not used, always set to 0.

Bit 3 - Decimal value 8. Not used, always set to 0.

Bit 4 - Decimal value 16. Message Available Queue Summary Message (MAV). The state of this bit indicates whether or not the output queue is empty. The MAV summary message is true when the output queue is not empty. This message is used to synchronize information exchange with the controller. The controller can, for example, send a query command to the device and then wait for MAV to become true. If an application program begins a read operation of the output queue without first checking for MAV, all system bus activity is held up until the device responds.

Bit 5 - Decimal value 32. Standard Event Status Bit (ESB) Summary Message. This bit indicates whether or not one or more of the enabled ESB events have occurred since the last reading or clearing of the Standard Event Status Register.

Bit 6 - Decimal value 64. Master Summary Status (MSS)/Request Service (RQS) Bit. This bit indicates if the device has at least one condition to request service. The MSS bit is not part of the IEEE-STD-488.1 status byte and will not be sent in response to a serial poll. However, the RQS bit, if set, will be sent in response to a serial poll.
Bit 7 - Decimal value 128. Not used, always set to 0.

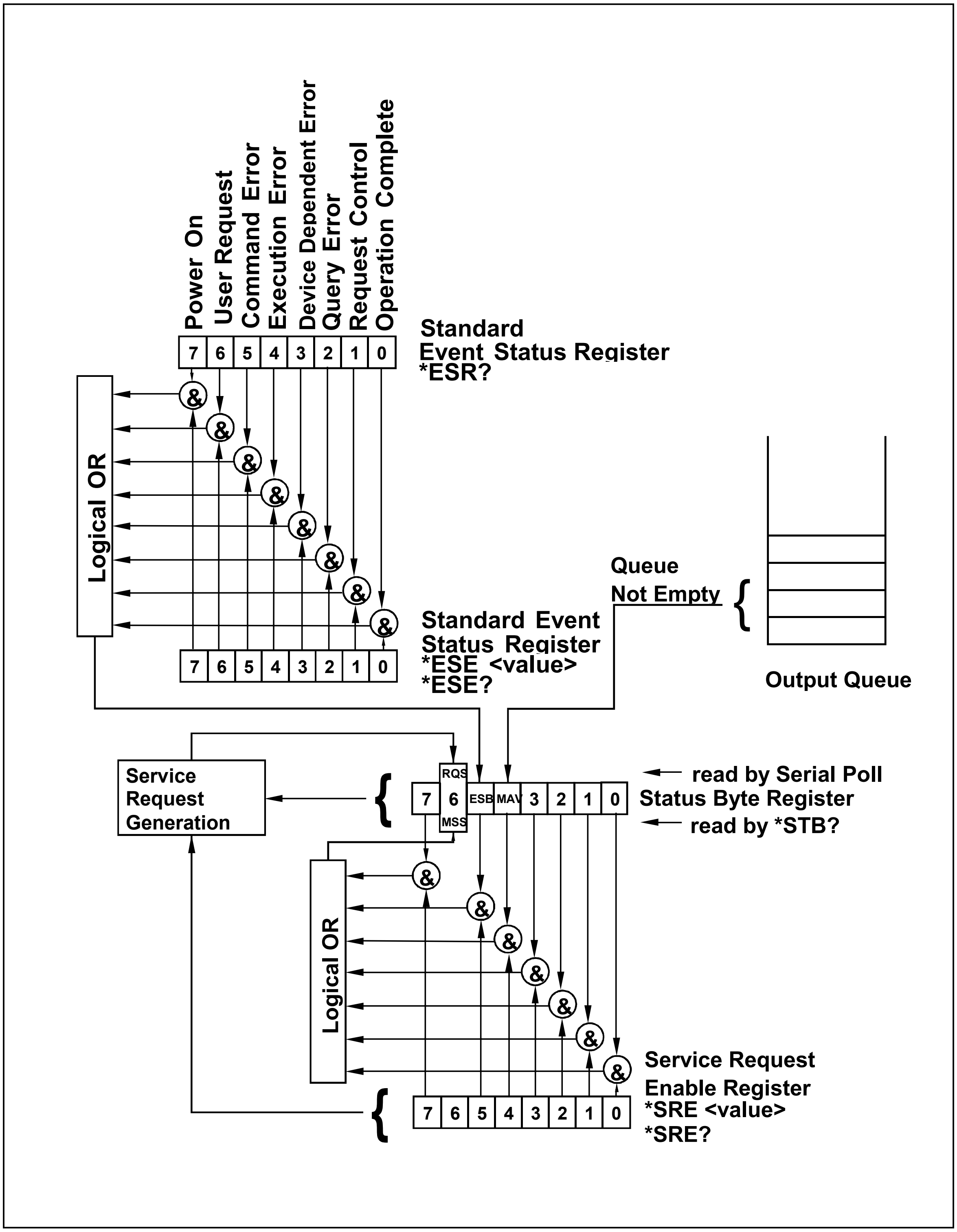


Figure 4-61. SCPI Status Registers

Reading the Status Byte Register

The Status Byte summary register can be read with the *STB? common query. The *STB? common query causes the generator to send the contents of the Status Byte register and the MSS (Master Summary Status) summary message as a single <NR1 Numeric Response Message> element. The response represents the sum of the binary-weighted values of the Status Byte Register. The *STB? common query does not alter the status byte.

Clearing the Status Byte Register

Removing the reasons for service from Auxiliary Status registers can clear the entire Status Byte register. Sending the *CLS command to the device after a SCPI command terminator and before a Query clears the Standard Event Status Register and clears the output queue of any unread messages. With the output queue empty, the MAV summary message is set to FALSE. Methods of clearing other auxiliary status registers are discussed in the following paragraphs.

Service Request Enable Register (SRE)

The Service Request enable register is an 8-bit register that enables corresponding summary messages in the Status Byte Register. Thus, the application programmer can select reasons for the generator to issue a service request by altering the contents of the Service Request Enable Register.

The Service Request Enable Register is read with the *SRE? common query. The response to this query is a number that represents the sum of the binary-weighted value of the Service Request Enable Register. The value of the unused bit 6 is always zero.

The Service Request Enable Register is written using the *SRE command followed by a decimal value representing the bit values of the Register. A bit value of 1 indicates an enabled condition. Consequently, a bit value of zero indicates a disabled condition. The Service Request Enable Register is cleared by sending *SRE0. The generator always ignores the value of bit 6. Summary of *SRE commands is given in the following.

***SRE0** - Clears all bits in the register.

***SRE1** - Not used.

***SRE2** - Not used.

***SRE4** - Not used.

***SRE8** - Not used.

***SRE16** - Service request on MAV.

***SRE32** - Service request on ESB summary bit.

***SRE128** - Not used.

Standard Event Status Register (ESR)

The Standard Event Status Register reports status for special applications. The 8 bits of the ESR have been defined by the IEEE-STD-488.2 as specific conditions, which can be monitored and reported back to the user upon request. The Standard Event Status Register is destructively read with the *ESR? common query. The Standard Event Status Register is cleared with a *CLS common command, with a power-on and when read by *ESR?.

The arrangement of the various bits within the register is firm and is required by all GPIB instruments that implement the IEEE-STD-488.2. Description of the various bits is given in the following:

Bit 0 - Operation Complete. Generated in response to the *OPC command. It indicates that the device has completed all selected and pending operations and is ready for a new command.

Bit 1 - Request Control. This bit operation is disabled on the Model PXA125.

Bit 2 - Query Error. This bit indicates that an attempt is being made to read data from the output queue when no output is either present or pending.

Bit 3 - Device Dependent Error. This bit is set when an error in a device function occurs. For example, the following command will cause a DDE error:

```
VOLTage 5;:VOLTage:OFFSet 2
```

Both of the above parameters are legal and within the specified limits, however, the generator is unable to generate such amplitude and offset combination.

Bit 4 - Execution Error. This bit is generated if the parameter following the command is outside of the legal input range of the generator.

Bit 5 – Command Error. This bit indicates the generator received a command that was a syntax error or a command that the device does not implement.

Bit 6 - User Request. This event bit indicates that one of a set of local controls had been activated. This event bit occurs regardless of the remote or local state of the device.

Bit 7 - Power On. This bit indicates that the device's power source was cycled since the last time the register was read.

Standard Event Status Enable Register (ESE)

The Standard Event Status Enable Register allows one or more events in the Standard Event Status Register to be reflected in the ESB summary message bit. The Standard Event Status Enable Register is an 8-bit register that enables corresponding summary messages in the Standard Event Status Register. Thus, the application programmer can select reasons for the generator to issue an ESB summary message bit by altering the contents of the ESE Register.

The Standard Event Status Enable Register is read with the *ESE? Common query. The response to this query is a number that represents the sum of the binary-weighted value of the Standard Event Status Enable Register.

The Standard Event Status Enable Register is written using the *ESE command followed by a decimal value representing the bit values of the Register. A bit value one indicates an enabled condition. Consequently, a bit value of zero indicates a disabled condition. The Standard Event Status Enable Register is cleared by setting *ESE0. Summary of *ESE messages is given in the following.

***ESE0** – No mask. Clears all bits in the register.
***ESE1** – ESB on Operation Complete.
***ESE2** – ESB on Request Control.
***ESE4** – ESB on Query Error.
***ESE8** – ESB on Device Dependent Error.
***ESE16** – ESB on Execution Error.
***ESE32** – ESB on Command Error.
***ESE64** – ESB on User Request.
***ESE128** – ESB Power on.

Error Messages

In general, whenever the PXA125 receives an invalid SCPI command, it automatically generates an error. Errors are stored in a special error queue and may be retrieved from this buffer one at a time. Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. When you have read all errors from the queue, the generator responds with a 0,"No error" message.

If more than 30 errors have occurred, the last error stored in the queue is replaced with -350, “Queue Overflow”. No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the generator responds with 0,"No error".

The error queue is cleared when power has been shut off or after a *CLS command has been executed. The *RST command does not clear the error queue. Use the following command to read the error queue:

```
SYSTem:ERRor?
```

Errors have the following format (the error string may contain up to 80 characters):

-102,"Syntax error"

A complete listing of the errors that can be detected by the generator is given below.

-100,"Command error". When the generator cannot detect more specific errors, this is the generic syntax error used.

-101,"Invalid Character". A syntactic element contains a character, which is invalid for that type.

-102,"Syntax error". Invalid syntax found in the command string.

-103,"Invalid separator". An invalid separator was found in the command string. A comma may have been used instead of a colon or a semicolon. In some cases where the generator cannot detect a specific separator, it may return error -100 instead of this error.

-104,"Data type error". The parser recognized a data element different than allowed.

-108,"Parameter not allowed". More parameters were received than expected for the header.

-109,"Missing parameter". Too few parameters were received for the command. One or more parameters that were required for the command were omitted.

-128."Numeric data not allowed". A legal numeric data element was received, but the instrument does not accept one in this position.

-131,"Invalid suffix". A suffix was incorrectly specified for a numeric parameter. The suffix may have been misspelled.

-148,"Character data not allowed". A character data element was encountered where prohibited by the instrument.

-200,"Execution error". This is the generic syntax error for the instrument when it cannot detect more specific errors. Execution error as defined in IEEE-488.2 has occurred.

-221,"Setting conflict". Two conflicting parameters were received which cannot be executed without generating an error. Listed below are events causing setting conflicts.

1. Sum of pulse or ramp parameters is more than 100. Corrective action: Change parameters to correct the problem.

2. ampl/2 + |offset| is more than 4. Corrective action: Reduce offset to 0, then change amplitude-offset values to correct the problem.

3. Activating filters when the PXA125 is set to output the built-in sine waveform, or activating the built-in sine waveform when one of the PXA125 filters is turned on. Corrective action: If in sine, select another function and activate the filter(s).

4. Activating burst mode when the PXA125 is set to sequence mode, or activating sequence mode when the PXA125 is set to burst mode. Corrective action: Remove the PXA125 from burst or sequence and then selected the desired mode.

- Changing operating mode from triggered to continuous when the PXA125 is set to single sequence advance, or changing the operating mode from continuous to triggered when the PXA125 is set to automatic sequence advance mode. Corrective action: Observe the PXA125 advance mode while setting sequence advance.

- Turning on FM, FSK, SWP, or Arm when one function is already on.

- Selecting TTLTrig line as a source advance when it has already been selected as an output.

-222,”Data out of range”. Parameter data, which followed a specific header, could not be used because its value is outside the valid range defined by the generator.

-224,”Illegal parameter value”. A discrete parameter was received which was not a valid choice for the command. An invalid parameter choice may have been used.

-300,”Device-specific-error”. This is the generic device-dependent error for the instrument when it cannot detect more specific errors. A device- specific error as defined in IEEE-488.2 has occurred.

-311,”Memory error”. Indicates that an error was detected in the instrument’s memory.

-350,”Queue Overflow”. The error queue is full because more than 30 errors have occurred. No additional errors are stored until the errors from the queue are removed. The error queue is cleared when power has been shut off, or after a *CLS command has been executed.

-410,”Query INTERRUPTED”. A command was received which sends data to the output buffer, but the output buffer contained data from a previous command (the previous data is not overwritten). The output buffer is cleared when power is shut off or after a device clear has been executed.

This page intentionally left blank

SPECIFICATIONS

Output Channels 1

Multiple Instrument synchronization

Description Multiple instruments can be connected together and synchronized to provide multi-channel synchronization.

Sample Clock

Source From Master card to slave boards through the local bus

Range and Resolution Same as Sample Clock but limited to 80 Ms/s

Initial Skew <20 ns to the first master; 20 ns cumulative to additional slaves

Phase Offset Between Instruments Programmable from 0 to n points

SAMPLE CLOCK

Range 100 mS/s to 125 MS/s

Resolution 10 digits limited by 1 µHz

Accuracy Same as reference

Stability Same as reference

Reference

Internal System clock, ≥0.01% (100 ppm) initial tolerance over a 19°C to 29°C temperature range; 100ppm/°C below 19°C and above 29°C; 100ppm/year aging rate

TCXO ≥0.0001% (1 ppm) initial tolerance over a 19°C to 29°C temperature range; 1ppm/°C below 19°C and above 29°C; 1ppm/year aging rate

External 10 MHz TTL, 50% ±2% duty cycle

SAMPLE CLOCK MODULATION

FM – Built-in Standard Waveforms

Description Sample clock can be frequency modulated by internal waveforms that are resident in internal memory (fixed waveforms).

Modulation Source Internal sine square, triangle and ramp

Modulation Frequency Range 2 mHz to 100 KHz

Resolution 10 digits

Accuracy 0.1%

Peak Frequency Deviation DC to 50 MHz

Advance Automatic, triggered, gated or software command

Marker

Output and Level Same as SYNC output.

Position Fixed at carrier frequency

FM – Downloaded Arbitrary Waveforms

Description Sample clock can be frequency modulate arbitrary waveforms that are downloaded by the user

Modulation Source User waveform, any shape, 10 to 20000 waveform points

Modulation Sample Clock Range 1 mS/s to 2 MS/s

Resolution 7 digits

Accuracy 0.1%

Peak Sample Clock Deviation DC to 125 MHz

Advance Automatic, triggered, gated or software command

Marker

Output and Level Same as SYNC output.

Position Programmable for selected sample clock frequency

Waveform Download Rate 5 Meg points per second

FSK

Description Current segment is sampled continuously. TTL low-level programs carrier sample clock, TTL high level programs shifted sample clock frequency. Sample clock changes coherently between frequencies. FSK operates on arbitrary waveforms only.

Carrier Sample Clock Range 100 mS/s to 125 MS/s

FSK Stimuli

External Front-panel Trigger input BNC. Low level = carrier sample clock; High level = hop frequency

Frequency Range From 10 MHz to DC

Internal Same as internal trigger range

FSK Delay Minimum 1 waveform cycle + 50 ns

Ramped FSK

Description Same as FSK except carrier sample clock ramps to shifted frequency at a rate defined by the ramp time parameter. TTL low-level programs carrier sample clock, TTL high level programs shifted frequency.

Ramp Time Range 10 µs to 1 s, 3 digits, ±0.1%

Sweep

Description Sample Clock sweeps continuously from start to stop, at a rate defined by the sweep time. More complex sweep modes and types can be generated using the FM mode in conjunction with the FM composer program.

Type	Linear or logarithmic
Direction	Up or down, depending on the start and stop setting
Range	100 mS/s to 125 MS/s
Time	1 ms to 1000 s, 7 digits, ±0.1%

Advance Automatic, triggered, gated or software command

Marker

Output and Level Same as SYNC output.

Position Programmable for selected frequency

OPERATING MODES

Normal Continuous waveform is generated

Triggered Each input cycle generates a single output cycle.

Gated External signal enables generator. First output cycle synchronous with the active slope of the triggering signal. Last cycle of output waveform always completed

External Burst Preset number of up to 128K cycles stimulated by an internal, or external trigger. This mode is not available in Sequence mode.

Trigger Sources

External

Input Front panel BNC

Level TTL

Slope Sensitivity Positive or negative, programmable

Frequency 5 MHz to DC

Internal

Range 100 mHz to 2 MHz

Resolution 7 digits

Accuracy 0.01%

Software SCPI command

Backplane TTLTrig0 through TTLTrig7; STAR

System Delay

Trigger to waveform output 1 Sample Clock+120 ns

STANDARD WAVEFORMS LIBRARY

Waveforms Sine, Triangle, Square, Pulse, Ramp, Sinc, Gaussian Pulse, Exponential decay/Rise Pulse, Noise, DC

Frequency Range Waveform dependent

ARBITRARY WAVEFORMS

Waveform Memory 2 Meg points

Memory Segmentation

Number of Segments 1 to 4096

Min Segment Size 16 points

Memory Interleave 4 (All trace lengths must be multiples of 4)

Vertical Resolution 14 bits (16,384 points)

Waveform Download Rate 5 Meg points per second

Sine Wave Performance

Description Sine wave performance is measured using the maximum sample clock rate, at 5 Vp-p, the maximum amplitude resolution of the DAC (14-bit) and without filters. Spectrum analyzer settings: Start frequency = 0 MHz; stop frequency = 50 MHz (Nyquist frequency)

Sinewave Total Harmonic Distortion 0.05% to 100 KHz

Harmonics and non-related spurious < -55dBc for carrier frequencies ≤1MHz
< -40dBc for carrier frequencies ≤5MHz

< -35dBc for carrier frequencies ≤10MHz

< -22dBc for carrier frequencies ≤50MHz

SEQUENCED ARBITRARY WAVEFORMS

Operation Permits division of the memory bank into smaller segments. Segments may be linked, and repeated in user-selectable fashion to generate extremely long waveforms.

Advance Modes

Automatic Sequence Advance No triggers required to step from one segment to the next. Sequence is repeated continuously through a pre-programmed sequence list (table)

Stepped Sequence Advance Current segment is sampled continuously, external trigger advances to next programmed segment. Control input is TRIG IN connector.

Single Sequence Advance Current segment is sampled to the end of the segment including repeats and idles there. Next trigger advances to next segment. Control input is TRIG IN connector.

Mixed Sequence Advance Each step of a sequence can be programmed to advance either a) automatically (Automatic Sequence Advance), or b) with a trigger (Stepped Sequence Advance)

Advance Source External, Internal, or soft trigger

Sequencer steps From 1 to 4096

Segment loops From 1 to 128 K

OUTPUTS

Waveform Output

Connector	Front panel BNC
Stand-by	Output Off or Normal
Impedance	50Ω, ±1%
Protection	Protected against temporary short to case ground

⚠ Amplitude

Range	80 mV to 8 Vp-p, into 50Ω; Double into open circuit
Resolution	3.5 digits

Accuracy (1 KHz)

800 mV to 8 Vp-p	±(1% + 10 mV)
80 mV to 799.9 mVp-p	±(1% + 1 mV)

Offset

Offset	Offset is attenuated with amplitude
Range	0 to ±3.6 V, amplitude dependent
Resolution	22 mV

Accuracy

±4 V window	±(1% of reading + 1% of amplitude + 2 mV)
±400 mV window	±(1% of reading + 1% of amplitude + 200 µV)

Filters 25MHz and 50 MHz, 7-pole elliptic

Square Wave, Pulse

Rise/Fall time	<10 ns, 10% to 90% of amplitude
Aberration	<5%, ±10 mV

SYNC/Marker Output

Description Provides dual functionality. All functions and modes, this output generates sync pulse, which is synchronous with the output waveform. In FM and sweep modes only, this output generates a marker having properties similar to the sync pulse output. The SYNC pulse can be routed to the PXI backplane to one or more of the TTL Trigger lines, thus providing trigger outputs to other modules. Front and backplane outputs are available simultaneously.

Outputs

Front panel SYNC Output BNC connector

PXI Backplane TTLTrig 0 through 7, programmable

STAR 0 through 7, drive slots 3 through 10 when module is plugged into slot 2 only. No individual control for each star line, all are either on, or off.

Validation

BIT Generates a sync signal each time a point within a specific wave segment is accessed.

LCOM (Loop Complete) Generates a single SYNC signal. Starts at a beginning of a sequence and ends at the end of the sequence.

Impedance 50Ω, ±1%

Level >2 V into 50Ω, 4V nominal into 10KΩ

Protection Protected against temporary short to case ground

Position Point 0 to n, Programmable with 4-point resolution

Width Programmable with 4 points resolution

Sine Output

Description An output that is directly derived from the sample clock generator and has an output frequency equal to the programmed sample clock frequency. This output generates sine waveform that corresponds to sample clock setting, including modulated waveform, such as FM, sweep and FSK

Connector Front panel SMB

Impedance 50Ω, ±1%

⚠ Level 1 V into 50Ω

Protection Protected against temporary short to case ground

Source Sample clock frequency

Frequency Range and Resolution Same as Sample clock

Flatness	-3dB at 100 MHz
Total Harmonic Distortion	0.3% to 100 KHz
	-55dBc to 1 MHz
Harmonics and non-related spurious	-45dBc to 10 MHz
	-35dBc to 100 MHz

INPUTS

TRIG Input

Connector	Front panel BNC
Impedance	10 KΩ, ±5%
Threshold Level	TTL
⚠ Damage Level	±10 V
Min Pulse Width	20 ns
Slope	Positive or negative going edge.

10 MHz Reference Input

Connector	Front panel SMB
Impedance	10 KΩ, ±5%
Threshold Level	TTL
⚠ Damage Level	±10 V
Duty Cycle	50%, ±5%

GENERAL

Power Requirements	10W maximum
Current Consumption	+5 V - 30 mA
	+12 V - 200 mA

	-12 V - 200 mA
	+3.3 V - 1.4 A
Dimensions	Single width, 3U high
Environmental	
Temperature (Operating)	0 to 50°C
Temperature (Non-Operating)	-40 to 71°C
Humidity (non-condensing)	5 to 95% RH at or below 30°C
	Upper limit derates to:
	75% RH above 30°C and
	45% RH above 40°C
Certifications	CE Marked
EMC	Conforms to EN 61326-1:1998
Safety	Conforms to EN 61010-1:2001
Workmanship Standards	Conforms to IPC-A-610D

This page intentionally left blank